NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MOBILITY AND CLOUD: OPERATING IN INTERMITTENT, AUSTERE NETWORK CONDITIONS

by

Toon Joo Wee
Yu Xian Eldine Ling

September 2014

Thesis Advisor:	Gurminder Singh
Co-Advisor:	Man-Tak Shing

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2014	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
MOBILITY AND CLOUD: OPERATING IN INTERMITTENT, AUSTERE NETWORK CONDITIONS

5. FUNDING NUMBERS

6. AUTHOR(S) Toon Joo Wee and Yu Xian Eldine Ling

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Cloud computing is emerging as the mainstream platform for a range of on-demand applications, services, and infrastructure. Before the benefits of cloud computing are realized, several technology challenges must be addressed. Operating in intermittent and austere network conditions is one of such challenges, which navy ships face when communicating with land-based cloud computing environments.

Given limited bandwidth and intermittent connectivity of satellite connections, mechanisms are needed to support data requirements of navy ships. Data caching and pre-fetching can be useful in such conditions. Caches act as temporary local storage and can satisfy data requests readily, if requested data is available. They promote faster response time and reduce bandwidth utilization. Cloudlets have the capability to bring the cloud closer to the users when they are deployed, as they extend the reachability of cloud-based servers to the users. In this research, we study the application of these two mitigating strategies in detail and evaluate their performance through modeling and simulation. Results from our simulations have suggested a positive impact. Caches and cloudlets as part of the shipboard architecture produce better performance in data communications. Most importantly, the strategies promote operations continuity for a naval force under disconnected, intermittent, and limited network environments.

14. SUBJECT TERMS
Cloud computing, cloudlet, mobility, cache

15. NUMBER OF PAGES
111

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

MOBILITY AND CLOUD: OPERATING IN INTERMITTENT, AUSTERE NETWORK CONDITIONS

Toon Joo Wee
Civilian, DSO National Laboratories, Singapore
B.Eng., Nanyang Technological University, 2006

Yu Xian Eldine Ling
Civilian, Defence Science & Technology Agency, Singapore
B.Eng., Nanyang Technological University, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2014

Authors: Toon Joo Wee
Yu Xian Eldine Ling

Approved by: Gurminder Singh
Thesis Advisor

Man-Tak Shing
Co-Advisor

Peter J. Denning, Ph.D.
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Cloud computing is emerging as the mainstream platform for a range of on-demand applications, services, and infrastructure. Before the benefits of cloud computing are realized, several technology challenges must be addressed. Operating in intermittent and austere network conditions is one of such challenges, which navy ships face when communicating with land-based cloud computing environments.

Given limited bandwidth and intermittent connectivity of satellite connections, mechanisms are needed to support data requirements of navy ships. Data caching and pre-fetching can be useful in such conditions. Caches act as temporary local storage and can satisfy data requests readily, if requested data is available. They promote faster response time and reduce bandwidth utilization. Cloudlets have the capability to bring the cloud closer to the users when they are deployed, as they extend the reachability of cloud-based servers to the users. In this research, we study the application of these two mitigating strategies in detail and evaluate their performance through modeling and simulation. Results from our simulations have suggested a positive impact. Caches and cloudlets as part of the shipboard architecture produce better performance in data communications. Most importantly, the strategies promote operations continuity for a naval force under disconnected, intermittent, and limited network environments.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
  A. THESIS OBJECTIVES 1
  B. RELEVANCE TO THE U.S. NAVY 1
  C. THESIS OUTLINE 2

II. BACKGROUND 5
  A. CURRENT COMMUNICATION TECHNOLOGIES 5
    1. Tactical Data Links 5
    2. Link-16 6
    3. Link-11 6
    4. Joint Range Extension Application Protocol 7
    5. Global Broadcast Service 7
    6. Comparison of Data Rates 8
  B. CURRENT SHIPBOARD DATA USAGE 8
    1. Command and Control Data 8
    2. Positioning Navigation and Timing Data 9
    3. Meteorological and Oceanographic Data 10
  C. RELATED WORKS AND TECHNOLOGIES 11
    1. Cloud Computing and Mobility 11
    2. Tactical Cloud Computing 12
    3. Mitigation Strategies 12
    4. Pre-fetching 13
    5. Related Work 14
    6. Cloudlets 15
      a. Advantages of Cloudlets 17
      b. Related Work 18

III. EXISTING NETWORKS AND PROPOSED MITIGATIONS 23
  A. APPLICATION OF C2 23
    1. C2 OODA Loop 23
    2. Data Profile and Characteristics 25
    3. QoS Requirements 26
    4. Communication Requirements 27
  B. SATELLITE COMMUNICATIONS 28
    1. Commercial Satellites as Supplement to Military SATCOM 28
    2. INMARSAT 29
  C. CLOUD RESPONSE TIME 30
    1. Challenges of Meeting Response Time Requirements 30
    2. Response Times of Amazon EC2 Instances 32
    3. Speed Testing on Amazon EC2 and S3 across Regions 33
    4. Using Amazon CloudFront To Improve Response Time 34
  D. SCENARIOS ANALYSIS 35
    1. Peacetime Use Case 35

2. Piracy Interdiction Use Case 38

E. PROPOSED MITIGATIONS 40

1. Local Caches 41

2. Deploying Cloudlet in the Architecture 42

IV. MODELING AND SIMULATION OF NAVAL COMMUNICATIONS AND CLOUD ARCHITECTURE USING QUALNET 43

A. SOFTWARE AND HARDWARE 43

1. Software Used 43

2. Hardware Used 44

B. APPROACH AND ASSUMPTIONS 45

C. MODEL DESCRIPTION 50

1. Base Case without Cache 50

2. Case 1: Modeling with Local Cache 52

3. Case 2: Modeling with Local Cache and Cloudlet 52

V. DISCUSSION OF RESULTS 57

A. BASE CASE WITHOUT CACHE 57

B. CASE 1: WITH LOCAL CACHE 59

1. Effect of Varying Cache Ratio 59

2. Effect of Varying the Number of Users 60

3. Effect of Varying Hit Ratio, P(hit) 61

C. CASE 2: WITH CLOUDLET 62

1. Faster Rate of Improvement in Performance with Higher P(hit) 63

2. High Data Load Converging to Fast Performance with High P(hit) 64

D. RECOMMENDATIONS 64

E. CHAPTER SUMMARY 65

VI. CONCLUSIONS AND FUTURE WORK 67

A. CONCLUSIONS 67

B. FUTURE WORK 68

APPENDIX A. DETAILED RESULTS FOR BASE CASE – WITHOUT CACHE 71

APPENDIX B. DETAILED RESULTS FOR CASE 1 – WITH CACHE 73

A. EFFECT OF VARYING CACHE RATIO ON AVERAGE RESPONSE TIME 73

B. EFFECT OF VARYING NUMBER OF USERS ON AVERAGE RESPONSE TIME 77

C. EFFECT OF VARYING HIT RATIO ON AVERAGE RESPONSE TIME 80

APPENDIX C. DETAILED RESULTS FOR CASE 2 – WITH CLOUDLET 83

A. EFFECT OF VARYING HIT RATIO ON AVERAGE RESPONSE TIME 83

LIST OF REFERENCES 87

INITIAL DISTRIBUTION LIST 91

LIST OF FIGURES

Figure 1. Representation of 3-tier hierarchy (from Li, Bu, Liu, & Xiao, 2013). 16
Figure 2. Proposed cloudlet-mesh architecture (from Khan, Wang, & Grecos, 2012). ..19
Figure 3. (a) Current mobile network topologies; (b) User-Cloud communication; (c) User-Cloudlet communication...21
Figure 4. Situation Awareness in context of OODA loop...24
Figure 5. Response time using Skype to stream CMG video...31
Figure 6. Response time using Citrix as CMD application..31
Figure 7. Response time of Small, Large, and Extra Large (H-CPU) instances. 33
Figure 8. Illustration of an edge cache nearer to end users. 35
Figure 9. Operational view: Maritime patrol operation. 37
Figure 10. Brief summary of data flow. 38
Figure 11. Operational view: Piracy interdiction operation..40
Figure 12. Deploying local cache on platform. 41
Figure 13. One node acting as a cloudlet. 42
Figure 14. Monthly Consumption Figures (per individual user) – North America, Fixed Access (from Sandvine, 2014)...46
Figure 15. Peak Period Aggregate Traffic Consumption – North America, Fixed Access (from Sandvine, 2014)...47
Figure 16. Hit ratio against cache size (from Beaumont, 2000).......................................49
Figure 17. QualNet Model for base case without cache..51
Figure 18. QualNet Model for base case with local cache and cloudlet. 54
Figure 19. Flow charts visually summarizing the three cases..56
Figure 20. Average response time plots for base case (with no cache)............................58
Figure 21. Average response time plots with (a) ten users, (b) 60 users..........................59
Figure 22. Varying the number of users with (a) 0.3, (b) 0.7 cache ratios.......................60
Figure 23. Varying P(hit) with (a) 0.3 (b) 0.7 cache ratios. 61
Figure 24. Average response time (with 0.3 Cache Ratio) (a) *without* cloudlet, (b) *with* cloudlet...62
Figure 25. Average response time (with 0.7 Cache Ratio) (a) *without* cloudlet, (b) *with* cloudlet...63
Figure 26. Fixed at 10 users. 73
Figure 27. Fixed at 20 users. 74
Figure 28. Fixed at 30 users. 74
Figure 29. Fixed at 40 users. 75
Figure 30. Fixed at 50 users. 75
Figure 31. Fixed at 60 users. 76
Figure 32. With 0.3 cache ratio. 77
Figure 33. With 0.4 cache ratio. 77
Figure 34. With 0.5 cache ratio. 78
Figure 35. With 0.6 cache ratio. 78
Figure 36. With 0.7 cache ratio. 79
Figure 37. With 0.3 hit ratio 80

Figure 38. With 0.4 hit ratio. 80
Figure 39. With 0.5 hit ratio. 81
Figure 40. With 0.6 hit ratio. 81
Figure 41. With 0.7 hit ratio. 82
Figure 42. With 0.3 cache ratio. 83
Figure 43. With 0.4 cache ratio. 84
Figure 44. With 0.5 cache ratio. 84
Figure 45. With 0.6 cache ratio. 85
Figure 46. With 0.7 cache ratio. 85

LIST OF TABLES

Table 1. Comparison of data rates of various TDL...8
Table 2. Key differences between cloudlets and clouds. 17
Table 3. Data categories. 25
Table 4. Examples of C2 data. 26
Table 5. Currently available commercial communication techniques (from Bekkadal, 2010). 28
Table 6. Response Time Requirement for CMG and CMD. 30
Table 7. Attributes of Amazon EC2 instances. 32
Table 8. Ten MB upload test across different regions. 34
Table 9. Average Response Time for base case (*without* cache and cloudlet). 71

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AIS	Automatic Identification System
AMAT	Average Memory Access Time
AO	area of operations
AP	access points
API	application programming interface
AWS	Amazon Web Services
BLOS	beyond line-of-sight
C2	command and control
CCI	Critical Contacts of Interest
CDN	Content Distribution Network
CMD	Cloud Mobile Desktop
CMG	Cloud Mobile Gaming
COP	Common Operating Picture
CTF	Carrier Task Force
DBS	Direct Broadcast Satellite
DDG	destroyer
DIL	disconnected, intermittent and limited
DSC	Digital Selective Calling
EC2	Elastic Compute Cloud
ECM	electronic countermeasure
EMIO	Expanded Maritime Interdiction Operations
ENDA	Embracing Network Inconsistency for Dynamic Application
EPIRB	Emergency Position Indicating Radiobeacon
FFG	frigates
GB	gigabyte
GBS	Global Broadcast Service
GPS	Global Positioning System
HADR	Humanitarian Assistance/Disaster Response
HF	high frequency
H-CPU	High-Central Computing Unit

IAAS	Infrastructure as a Service
IDE	Integrated Development Environment
IPO	International Program Office
ISR	Intelligence, Surveillance and Reconnaissance
JREAP	Joint Range Extension Application Protocol
JTIDS	Joint Tactical Information Distribution System
KBPS	Kilobits per second
LOS	line-of-sight
MB	Megabyte
MBPS	Megabits per second
METOC	Meteorological and Oceanographic
MIDS	Multifunctional Information Distributed System
MOC	Maritime Operations Centre
MTC2	Maritime Tactical Command and Control
NAVSSI	Navigation Sensor System Interface
NAVTEX	Navigational Telex
NIST	National Institute of Standards and Technology
OMNeT++	Objective Modular Network Testbed in C++
OODA	Observe, Orient, Decide and Act
OPNET	Optimized Network Engineering Tools
OV	Operational View
PNT	Positioning, Navigation and Timing
QoS	Quality of Service
RAM	Random Access Memory
S3	Simple Storage Service
SAR	Search and Rescue
SATCOM	satellite communications
SIPRNET	Secret Internet Protocol Router Network
SSAS	Ship Security Alert System
TACT HQ	Tactical Headquarter
TDL	Tactical Data Links
TDMA	Time-Division Multiple Access

TR	Total data volume
TTL	Time-To-Live
UAV	unmanned aerial vehicle
UHF	ultra-high frequency
USN	U.S. Navy
VSAT	Very Small Aperture Terminal
WAN	wide area network
XML	Extensible Mark-up Language

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Many people have helped us in this thesis, one way or another. We would like to specifically express our heartfelt appreciation to our thesis advisors, Professor Gurminder Singh and Professor Man-Tak Shing. Without their constant guidance, feedback, and suggestions, this thesis would not have gone so smoothly. We would also like to thank Professor John Gibson for his advice and sharing of his experience and expertise on modeling and simulation software. Our appreciation is also extended to Professor Weilian Su and Mr. Robert Broadston for their assistance and support on the QualNet software, and to CDR Robert Yee for the helpful discussion on naval communication.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Cloud computing has rapidly become the mainstream platform for a range of applications and services. For any large-scale, scalable, networked application system, the default starting point for hosting is the cloud, the foundations of which lie in distributed computing, networking, databases, load-balancing, and security. In most cloud-based systems, clients generate and/or consume information, and are connected to cloud-based servers over wired or wireless network connections. For mobile clients, this connection, by default, is a wireless connection. While cloud computing has brought about unprecedented sophistication in the mobile ecosystem, there are a number of issues that need to be addressed in order for the overall environment to be dependable.

A. THESIS OBJECTIVES

Our goal is to develop solutions to the challenges created by mobile-cloud interactions. These challenges arise due to wildly fluctuating wireless bandwidth, intermittent connectivity, and unreliable connectedness of mobile clients. Navy ships face the same issues when communicating with cloud-based servers. We plan to study existing mitigation strategies used in the commercial cloud industry, and also review the U.S. Navy command and control data. From there, we will propose relevant and possible strategies to overcome the challenges identified. Taking alignment to navy scenarios, verification of the proposed strategies is conducted and recommendations to the existing architecture will be made.

B. RELEVANCE TO THE U.S. NAVY

The scenarios and models developed in this thesis are useful for the analysis of existing and future data requirements on U.S. Navy ships. To provide a smooth user experience in fluctuating bandwidth environments, it is important to anticipate the application’s information requirements to pre-fetch and cache data that may be needed in the near future. In addition to intermittent connectivity issues, many wireless networks introduce additional delay due to excessive buffering. This can be a problem for both consumers as well as producers of information in the navy. The strategies proposed would identify and handle real-time, near real-time, and non-real-time data in the mobile-cloud scenarios. Benefits from the proposed strategies could also be extended to the other land-based platforms, as well as enterprise systems.

C. THESIS OUTLINE

In Chapter II, we present the current available communication technologies which support naval operations. We also review the different data usages by the U.S. Navy. In order to broaden our knowledge on existing technologies in the mobile cloud field, we explore related research works, focusing on areas which involve data structuring, hosting, and caching strategies for systems, as well as the implementation of cloudlets. Our findings can provide valuable insight for architecting shipboard systems and help deliver a smoother user experience.

In Chapter III, we discuss the existing environment in which the U.S. Navy platforms are operating and propose relevant strategies that will help to overcome challenges. We consider application characterization based on data type prioritization, which together with the consideration of the satellite communications (SATCOM) connectivity, will keep us align to realistic mission requirements and facilitate our thought process when proposing relevant strategies.

We seek to verify our proposed solutions in Chapter IV, where we conduct our experiments and modeling of the operational scenario using a communication network simulation application, QualNet. We wanted to model our operating environment as close to the actual naval environment as possible, but due to the security classification of the existing data set, we were only able to base our data usage on the actual Internet traffic in the public domain. Simple scenarios were created and relevant data input, which were extracted from our study of the data, and were executed against the models. The models demonstrate the advantages of the implementation of both the cache and cloudlet. We adopted an incremental phased approach in our testing where we first modeled a scenario without cache implementation. This forms the base case of our experiment. The next phase is to test the scenario with cache implementation and, thereafter, comparing the results against the base case's. The last phase is built on the second scenario, to include the implementation of cloudlets. Again, the results are compared with the previous scenarios.

Results and findings are discussed in Chapter V, and the analysis of the data has shown that the strategies proposed are able to support our claims in providing resolution to the identified problems in the current architecture. First, we are able to show that the implementation of caching can improve the response time of requests made by the users. Second, we are positive that the implementation of cloudlets can indeed bring the "cloud closer to the users." The results have demonstrated that the response time was shortened by at least 20 to 30 percent, which is a significant improvement to the case where all requests had to be made to the cloud server. This finding is based on the lowest cache hit ratio. Input parameters such as the cacheable ratio and the cache-hit ratio were varied accordingly to find their optimal use cases. There is also no unexpected result on the analysis of cloudlets and, in fact, performance on the connectivity is further improved. The increase in the number of connecting nodes does not affect sending requests and receiving replies, and the operation can still continue even in the absence of a direct connection to the satellite. In our case, even though the cloudlet node suffered an increase in data load, our experiment proves that it is still able to operate satisfactorily in a degraded mode. As compared to operating in a disconnected environment, this degraded performance provides a much better alternative.

Although theoretical results from our experiment have shown the feasibility of implementing caches and cloudlets, we would still need to carry out practical evaluations of the proposed strategies to further verify the effectiveness of the solutions. This could be part of the future work to be done. First, we need to validate the theoretical results by running the experiments again with actual data from the navy. Second, the scope of our thesis covers the case where there is only one cloudlet node. Moving forward, the scope of the test can be extended to cover the case where all the nodes can be cloudlet nodes. This can be achieved by tweaking the inputs and reworking the formulas used in the model. We foresee that this would be similar to an ad-hoc meshed topology, where all nodes can access other cloudlets first before establishing a remote connection back to the remote cloud. In addition, due to the limitation of the modeling software, we were not able to incorporate pre-fetching algorithms into our models. Initial research has indicated that pre-fetching can contribute to mitigating latency issues; perhaps this solution too, can be verified in subsequent research or experimentation.

II. BACKGROUND

The U.S. Navy (USN) and its coalition partners have become increasingly dependent on the availability of stable and robust ship-to-shore SATCOM to deliver network and application services (Lapic, 2013). While SATCOM has provided unprecedented support for military services, the communication supported by SATCOM is less than reliable in terms of its quality of service and connectivity. Because of this several services can be adversely affected. Command and control (C2) is an example application that will be greatly affected by intermittent communication.

Tactical situations increase the likelihood of a disconnected, intermittent, and low-bandwidth (DIL) environment while simultaneously increasing the need for an updated and synchronized common operational picture (COP) (Perkins, 2013). Existing C2 systems using event-based protocols to manage tracks may conserve bandwidth, but they do not guarantee a common operating picture in DIL environments. Without an updated COP among the involved parties, confusion can arise among them, and well-informed and sound decisions cannot be made.

For these reasons, many researchers have proposed architectures designed for DIL environments. Motivations for these efforts have derived from not only concerns about the risk that satellites can be jammed or even shot down during hostilities, but also concerns about the cost and availability of satellites world-wide and to all partners in potential coalitions even in peacetime.

A. CURRENT COMMUNICATION TECHNOLOGIES

The following subsections provide an introduction to the current communications that are applicable to our thesis study.

1. Tactical Data Links

Tactical data links (TDLs) provide a means to exchange information between different afloat entities and to access information from a remote source. The information can be textual information such as from intelligence reports or imagery and video from radar, sonar, unmanned aerial vehicle (UAV), etc. With effective communications, the information adds up to a better situational picture in a C2 context.

The following TDLs and their related protocols are relevant to our thesis (Grumman, 2013).

2. Link-16

Link-16 is a military tactical data exchange network used by the United States, NATO, and nations allowed by the Multifunctional Information Distributed System International Program Office (MIDS IPO).

With Link-16, military aircraft as well as ships and ground forces may exchange their tactical picture in near-real time. Link-16 also supports the exchange of text messages and imagery data, and provides two channels of digital voice.

Link-16 employs the Joint Tactical Information Distribution System (JTIDS) and MIDS data link interfaces. Link 16 is a frequency-hopping, jam-resistant, high-capacity data link. It is operating on the principle of Time-Division Multiple Access (TDMA), where 128 time slots per second are allocated among participating JTIDS units.

3. Link-11

Link-11 employs netted communication techniques and standard message formats for the exchange of digital information among airborne, land-based, and shipboard tactical data systems. Providing mutual exchange of information among net participants using high-frequency (HF) or ultra-high-frequency (UHF) radios, Link-11 is a half-duplex, netted, secure data link.

If operating on HF ground wave, it has beyond line of sight (BLOS) capability to a theoretical range of approximately 300 nautical miles (nm). The downside is that it is not electronic countermeasure (ECM)-resistant.

4. Joint Range Extension Application Protocol

The most common method of extending JTIDS range BLOS is the employment of airborne relays. This is not always feasible, however, due to the lack of airborne assets or conflicting mission requirements. Several means of employing satellite communications to extend the range of Link-16 are under development as part of the Joint Range Extension Application Protocol (JREAP) program. JREAP is an application protocol that enables the transmission of tactical data link messages over media that is not originally designed for TDLs. The communications which are supported include the following:

(1) Satellite Communication – JREAP-A

JREAP-A uses a token passing protocol over half-duplex communication channels to send and receive TDL messages. JREAP-A implements the full-stack header and uses a token passing protocol, where one unit is allocated a particular period of time to transmit data while all other units listen and receive the data.

(2) Point-to-Point – JREAP-B

JREAP-B is used in synchronous or asynchronous point-to-point communications. JREAP-B is commonly used with full-duplex serial data communications.

(3) IP Networks – JREAP-C

JREAP-C is an implementation of the JREAP protocol that transmits TDL messages over Internet Protocol networks such as Secret Internet Protocol Router Network (SIPRNET). JREAP-C differs from JREAP-A and JREAP-B by implementing the application header instead of the full-stack header. This is done because the error detection and correction as well as addressing are not necessary as they are handled by the lower layers of the stack. This permits fast and reliable transmission of messages over a network.

5. Global Broadcast Service

Global Broadcast Service (GBS) (Military.com, 2005) provides a worldwide, high capacity, one-way transmission means for a variety of data, imagery, and other information required to support joint forces. GBS capitalizes on the popular commercial direct broadcast satellite technology to provide critical information to warfighters. The GBS system is a space-based, high data-rate communications link for the flow of information from the United States or rear echelon locations to deployed forces.

The GBS system “pushes” a high volume of intelligence, weather, and other information to widely dispersed, low-cost receive terminals, similar to the set-top-box used with commercial Direct Broadcast Satellite (DBS). The system includes a capability for the users to request or “pull” specific pieces of information. These requests are processed by an information management center where each request is prioritized, the desired information requested is retrieved, and then scheduled for transmission.

6. Comparison of Data Rates

The raw data rates (in kilobits per second (kbps)) are shown in Table 1.

Table 1. Comparison of data rates of various TDL.

Link-16 JTIDS	Link-11	JREAP	GBS (one-way and shared)
26.88-107.52	1.09 or 1.8	Varies (depend on mission)	48,000

B. CURRENT SHIPBOARD DATA USAGE

The following subsections provide an introduction to the current shipboard data usage that is critical to the naval operations.

1. Command and Control Data

“Information Dominance” is a major factor to the success of any C2 system, especially in a Network Centric Warfare. The accuracy and availability of the required information directly affects the effectiveness of C2 operations. Therefore, we must understand the requirements of a C2 system.

Situational awareness is a vital ingredient of a C2 system. The quality of a commander's decision for the next course of action greatly depends on the accuracy and timeliness of situational awareness provided by the C2 system. We can imagine that a gap in the knowledge about a certain situation affects the decision greatly. For example, the intelligence on the red force whether mines are deployed in a region might be the deciding factor for the blue force to send in reinforcements or what kind of platform to traverse through that region.

To accomplish a sufficiently effective situational awareness, the required components might include:

- Present and future positioning of red force (this information can be obtained from sensors, human intelligence, signal intelligence, communications intelligence, image intelligence, or even open-source intelligence)
- Time of reported intel/event
- Updated maps

In order to automate some of the C2 functions, data fusion (Waltz, 1986) of the various sources of information (i.e., merging and making sense of the incoming information) is essential to providing an accurate and timely situational awareness picture to the commander. The order in which the information comes into the system might paint a completely different picture to the commander.

2. Positioning Navigation and Timing Data

Positioning Navigation and Timing (PNT) distribution systems are required to provide a common geospatial platform and temporal reference to military platforms. This data is pervasive and critical for military platforms, because it supports many targeting, situational awareness, communication, and weapon systems. Overall mission effectiveness is also highly dependent on PNT data (Osa, 2004). In this age of information and communication, the challenges faced by PNT systems lie in the areas of data distribution, and time and frequency synchronization. (Shaw, 2004)

Similar to tactical cloud computing, interoperability remains as the main challenge in the pervasiveness of PNT data in U.S. Navy systems. The Navigation Sensor System Interface (NAVSSI), being the primary source of PNT data, gathers inputs from multiple shipboard sensors and then distributes the resultant navigation, time, and frequency data to both internal and external systems for consumption.

Time criticality is the other important factor in distributing PNT data to naval systems. This is highlighted by (Shaw, 2004) as he discusses the criticality of PNT data used in targeting, weapons, and communication systems.

3. Meteorological and Oceanographic Data

Weather conditions in both the atmosphere and ocean can affect how the U.S. Navy carries out their operation. It is difficult to make an accurate prediction of the weather, and this impedes the naval forces from planning and executing their mission efficiently and effectively. Force structure composition, force movement prediction, personnel safety, estimation of capability performance, and war-fighting tactics are examples of what the adverse weather can impact. Hence, the need for meteorological and oceanographic (METOC) information is critical. Furthermore, the advances in technology have also introduced other challenges to obtaining accurate METOC data. These include:

- Changes to the support infrastructure of METOC.
- Complication of the forecast process due to proliferation of information technology systems.
- Addition of new METOC science and data sources, increasing the knowledge required and standards of operators.
- Overload of METOC data, causing the operator to be overwhelmed and affecting the ability to produce accurate forecast results.

In order to overcome these challenges, the U.S. Navy is composed of a hierarchy of support elements, each tasked with analyzing METOC data and converting them to useful METOC information that can aid naval forces in conducting their mission.

C. RELATED WORKS AND TECHNOLOGIES

The following subsections provide the discussion of the related works and technologies which help to build up the strategies that are proposed in Chapter IV.

1. Cloud Computing and Mobility

Cloud computing is rapidly becoming the mainstream platform for a range of networked applications and services, which are large scale and scalable. It is receiving a great deal of attention, both in publications and among users, spreading from home to government offices. Yet the definition of cloud computing is not clear (Huth, 2011). The meaning of the term “cloud computing” is rapidly evolving with the maturity of technology. The National Institute of Standards and Technology (NIST) defines cloud computing as “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” (Foster, 2010). In other words, we can consider cloud computing as a subscription-based service to obtain networked storage space and computer resources. For example, a traditional computer setup requires users to be physically with their data storage device; the cloud eliminates this constraint. The cloud provider can both own and house the hardware and software necessary to run a user’s home or business applications. This, as a matter of fact, enables mobility. The cloud makes it possible for users to access their information from anywhere at any time (Huth, 2011).

There are different types of clouds that users can subscribe to depending on their needs (Huth, 2011).

- Public Cloud - A public cloud is available to any user with an Internet connection.
- Private Cloud - A private cloud is available only to a pre-determined group of users.
- Community Cloud - A community cloud is available to two or more groups with similar requirements.
- Hybrid Cloud - A hybrid cloud is a combination of public, private, or community clouds.

2. Tactical Cloud Computing

Our research focuses on a different type of cloud, called the Tactical Cloud. The advent of net-centric warfare has allowed situational awareness to be shared among distributed forces that are connected in the cyber space. Tactical cloud computing has become a key enabler in distributing timely and accurate information to warfighters for effective decision making, and during an engagement. Cloud computing can provide the necessary globally distributed computing environment for the "edge entities" who are conducting military missions to "smart pull" information from ubiquitous sources anytime and anywhere (Foster, 2010; Michael, 2011).

3. Mitigation Strategies

In the commercial world, methodologies for how to overcome challenges have been developed to make their cloud applications a reality. For example, Pan (2012) discusses adaptive mobile video streaming and user behavior-oriented pre-fetching from the cloud. More recently, Liu (2013) has talked about the methodology to run richer applications on resource-constrained mobile devices by offloading a part of the computation to a powerful cloud. With monetary motivations and yet less stringent requirements, commercial cloud applications have grown at a fast rate.

Similar to other mobile systems, navy ships connected over wireless networks often face the issue of intermittent connectivity. In order to operate in an intermittent network environment, we will investigate architectures which enable smooth operation of ship-board systems. We expect to handle information needs of near-real-time and non-real-time systems, possibly extending to real-time systems as well.

We are exploring the following approaches which are commonly used to solve problems of mobile systems, applying them to a specific naval application/domain which can benefit from tactical cloud:

- To provide a smooth user experience in fluctuating bandwidth environments, one of the ways is to anticipate application's information requirements and pre-fetch data that may be needed in the near-future steps. This may parallel pre-fetching strategies used in disk drives and operating systems, but research is needed to explore these in the context of cloud and mobility.
- In addition to the connectivity issues, many wireless networks introduce additional delay due to excessive buffering. This can be a problem for both consumers as well as producers of information. We plan to investigate strategies to identify and handle real-time, near-real-time, and stored data in the mobile-cloud scenarios.

4. Pre-fetching

Pre-fetching (Cepeda, 2009), as used in computer architecture, means bringing data or instructions from memory into the cache before they are needed. When an application needs data that was pre-fetched, instead of waiting for the data from memory, the application can grab it from the cache and keep right on executing.

In mobile cloud computing, the pre-fetching technique is also used in a similar way. With improvement of technology on mobile devices, the demand for mobile cloud computing services is growing, which inevitably leads to issues such as latency. Pre-fetching and caching are popular techniques to mitigate latency issues. The pre-fetching technique reads and stores data objects that are not currently required but are predicted (statistically) to be needed in the near future. Essentially, the purpose of pre-fetching (Zhang, 2013) is to supplement the caching strategies. In mobile cloud computing, this cache can exist in a form of cloudlet which is further discussed later in this chapter.

5. Related Work

There are many challenges in implementing pre-fetching schemes. First, it is difficult to find which objects are related to the incoming request. Second, it is difficult to find an optimal pre-fetch rate. Too aggressive pre-fetch schemes may hurt overall performance due to memory shortages and wasted bandwidth. Lee (2009) proposes an adaptive web pre-fetch scheme in web cluster environments to support the new web framework by designing the Double Prediction by Partial-Match Scheme (DPS). Yet, this method loses the possibility to enhance the hit rate on the memory of the pre-fetch.

From a disk access perspective, Yoon (2010) proposed a two-level pre-fetching technique, which considers both file-level and block-level with sequential access patterns in pre-fetching. This algorithm demonstrates good performance on disk pre-fetching. However, it does not have a high efficiency at pre-fetching data in a cloud computing environment, because the patterns are over-fitted and many data objects are pre-fetched into the local data nodes, which puts a heavy burden on the network.

Kung (2010) presents some speculative pipelining solutions for compute cloud programming. They used speculative prefetching and computing to minimize execution delays in subsequent stages due to varying resource availability. However, decoding is required which dampens performance. Moreover, encoding requires more storage for overhead.

Li (2012) proposes a real-time data pre-fetching algorithm based on sequential pattern mining that could be used to hide the data access delay, which has become a leading factor that affects Quality of Service (QoS). These issues are solved by predicting the data object that the user will request and pre-fetching it to local data nodes. This reduces network overhead in data transmission and avoids excessive pre-fetching. The proposed algorithm directly pre-fetches data objects from remote data nodes to local data nodes' cache space. However, this algorithm does not consider the replacement and consistency strategy.

Wang (2013) discusses the mediocre service quality of video streaming services via mobile networks. Cloud-assisted adaptive mobile video streaming and socially aware pre-fetching have been proposed, which would enable videos to be efficiently stored in clouds for mobile users. The weakness of the solution is that energy and transmission costs are not considered in the usage profile.

Based on these previous works, it is shown that to overcome the issues related to latency and data access delays, pre-fetching can be used. However, the proposed pre-fetching methods have their own shortcomings.

The data pre-fetching techniques provide the best results when we have determined the data requirements. First, this approach can recognize the useful pre-fetches, like pre-fetch data that is more likely to be requested by the processor in the future. Second, the pre-fetches should be timely. The data should not be brought in too early before it is required in the future, nor should it be brought in too late. It is not going to be useful because some pre-fetching does not hide a significant portion of memory access latency. Finally, the pre-fetches should not be corrupting the cache, by displacing data that is going to be used with data that is pre-fetched but not used.

Mobile naval platforms should benefit from using pre-fetching, but there are other issues to consider with mobility, which adds new challenges to the pre-fetching problem. For example, we need to understand the specific characteristics that come with each type of network link.

6. Cloudlets

There has been a wide range of cloud applications since the introduction of cloud computing. These applications fall into categories like audio/video processing, image processing, GPS sharing, sensor data applications, multimedia search and streaming applications. These applications involve distributed computation and require clear fragmentation of data. The time taken to recombine these fragments of data must also be very small. In addition, in order for some of these resource-demanding applications to achieve real-time capabilities, they are required to access the back-end server which resides in the cloud. However, these clouds are typically located far away from the user, resulting in high wide area network (WAN) latency. As a result, it is unsuitable for the data of the application to be accessed in real-time.

To cope with this high latency, Satyanarayanan and his colleagues (Satyanarayanan, Bahl, Caceres, & Davies, 2009) introduced the concept of cloudlets: trusted, resource-rich computers in the near vicinity of the mobile user. Integrating distributed and localized cloudlets with quickly deployable, low-cost wireless mesh networks will facilitate low-latency and prompt access to data for cloud-based applications (Khan, Wang, & Grecos, 2012).

A cloudlet can be viewed as a “data center in a box” whose goal is to “bring the cloud closer” to the user (Satyanarayanan, n.d.). In Figure 1, the cloudlet tier lies in the middle of the 3-tier hierarchy: the mobile device—the cloudlet layer—the cloud.

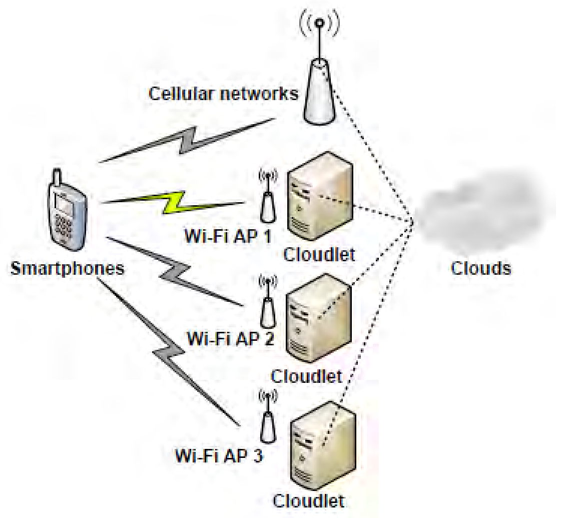


Figure 1. Representation of 3-tier hierarchy (from Li, Bu, Liu, & Xiao, 2013).

According to Satyanarayanan (n.d.), a cloudlet consists of the following attributes:

- **Comprised of soft state.** It may contain the cloud’s cached state and also buffer data originating from a mobile device to the cloud.

- **Self-managing**. Because cloudlets only hold soft states, they do not require much management.

- **Powerful, well-connected, and safe.** Cloudlets may be constantly connected to the Internet if serviced by a wired connection, resulting in a more guaranteed connectivity. Cloudlets can be full powered by a power outlet depending on the physical environment. A cloudlet could also possess enough computing power to offload resource-intensive computations from one or more mobile devices.

- **Close proximity.** It is logically proximate to the associated mobile devices. “Logical proximity” is defined as low end-to-end latency and high bandwidth (e.g., one-hop Wi-Fi).

- **Heterogeneous.** It aligns to the current cloud architecture and is able to be scaled from the current cloud. Base on their assigned roles, each cloudlet has its own specific function.

Table 2 summarizes some of the key differences between cloudlets and clouds (Satyanarayanan, Bahl, Caceres, & Davies, 2009).

Table 2. Key differences between cloudlets and clouds.

	Cloudlet	Cloud
State	Only soft state	Hard and soft state
Management	Self-managed; little to no professional attention	Professionally administered; 24/7 operator
Environment	“Data center in a box” at business premises	Machine room with power conditioning and cooling
Ownership	Decentralized ownership by local business	Centralized ownership by Yahoo, Amazon, etc.
Network	Local area network latency/bandwidth	Internet latency/bandwidth
Sharing	Few users at a time	Hundreds to thousands of users at a time

a. Advantages of Cloudlets

Cloudlets have emerged as a promising architecture to reduce communication delay. Being a resource-rich element that has good connectivity to the Internet and mobile devices, cloudlets allow low end-to-end latency to be achieved. This is analogous to Wi-Fi access points (AP) which are in close proximity to a user’s mobile device, allowing the mobile device to enjoy a higher signal strength and higher speed access to the Internet.

Due to such characteristics of the cloudlet technology, cloudlets are relevant for use in hostile environments. A hostile environment is attributed with short-term but high uncertainties. (Satyanarayanan, et al., 2013) Some examples of hostile environments include the theater of a military operation, an area recovering from a natural disaster, and even a developing country with a weak network infrastructure.

The physical proximity of cloudlets proves advantageous to serving such hostile environments. It is not realistic, however, to create a cloud in range of every mobile device. Cloudlets can bridge this proximity gap. Instead of relying on a cloud that is far away and being susceptible to poor connectivity, the cloudlet can provide a nearby and resource-rich alternative. This way, the requirement for a low latency, real-time, efficient bandwidth utilization can be achieved.

To further elaborate on the benefits of cloudlets, for example, consider a mobile device that is functioning as a thin client. This client is connected to the cloudlet and is accessing the applications within it at real-time. If there is no cloudlet available nearby, the mobile device can gracefully downgrade and work in a degraded mode, connecting to the farther remote cloud. If at any time, when the client detects an available cloudlet, it can regain its faster connectivity and achieves full functionality and performance. Basically, cloudlets are decentralized and distributed Internet infrastructure components whose computing resources can be leveraged by nearby mobile clients. In other words, a cloudlet resembles a "data center in a box" that "brings the cloud closer;" it is self-managing, requires little power, has Internet connectivity and access control for setup (Satyanarayanan, Bahl, Caceres, & Davies, 2009).

Other subtle benefits of having cloudlets include safe deployment in insecure areas such that tampering, loss, or destruction of cloudlets do not prove to be a major security issue. This is due to the content of cloudlets being in soft states only.

b. Related Work

Researchers have put in a lot of effort looking into how the advantages of cloudlets can be reaped, and one of the concepts that we can base our modeling on is the concept of an integrated Cloudlet-Mesh Architecture. The integrated architecture, as shown in Figure 2, is comprised of full-scale data centers (also known as the cloud), the cloudlets, and wireless meshed networks.

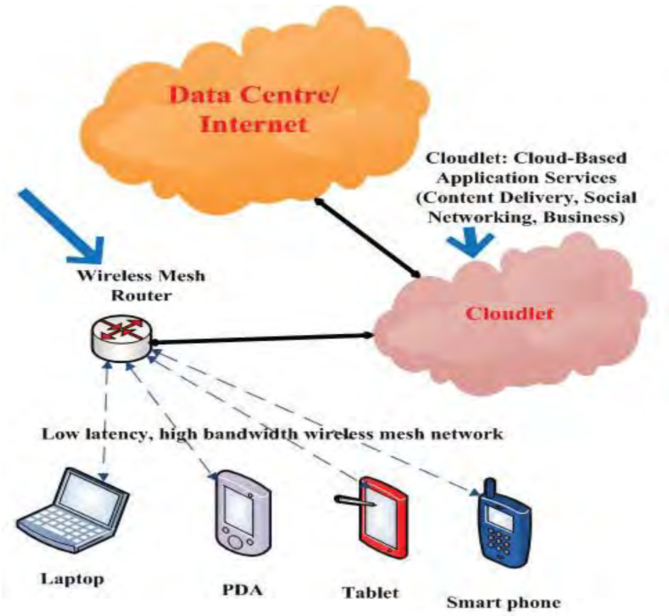

Data Centre/ Internet
Cloudlet: Cloud-Based Application Services (Content Delivery, Social Networking, Business)
Wireless Mesh Router
Cloudlet
Low latency, high bandwidth wireless mesh network
Laptop
PDA
Tablet
Smart phone

Figure 2. Proposed cloudlet-mesh architecture (from Khan, Wang, & Grecos, 2012).

Figure 2 is only a simplified representation of the architecture. However, it is proven that the architecture is scalable and is able to support the connection to a number of cloudlets, and each cloudlet can connect to one or more wireless meshed routers to provide a more robust and wider coverage. With the integrated architecture, users can roam around without having to be disconnected from the wireless router. Instead, more user sessions can be kept connected with the extended range and smoother transitions are guaranteed. This architecture not only explores the benefits of deploying distributed and localized cloudlets to bring cloud resources closer to end users, facilitating and improving the performance of resource-demanding and delay-sensitive applications, but it also benefits from the advantages of wireless mesh networking in terms of cost efficiency, instant deployment, self-organizing, and flexible access to the localized cloud services.

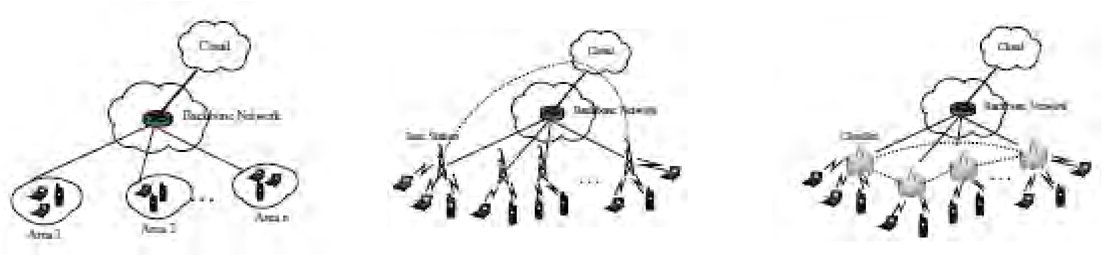
In another related research work, the solution Embracing Network Inconsistency for Dynamic Application Offloading in Mobile Cloud Computing, or ENDA, was proposed to help overcome network inconsistency. ENDA deploys a three-tier architecture, where components on each tier interact with one another, and utilizes user track prediction, real-time network performance, and server loads to optimize offloading decisions (Li, Bu, Liu, & Xiao, 2013).

Some advantages of ENDA that were highlighted in the literature include:

- Prediction of user tracks to estimate the motion and trajectory of the user. This is done by implementing a search algorithm to monitor users' traces in database servers on the cloud.
- Consideration of the user profile which lowers the chance of network performance downgrading and network disconnection.
- Implementation of management mechanisms for workload balancing among cloudlets, based on server loading and quality of network, and a fault-tolerant mechanism to handle inaccurate offloading decisions.
- Reduction of energy consumption on mobile devices by division of labor among the components in the 3-tier architecture. The complex computation is undertaken by the cloud and cloudlets while the mobile devices are responsible for transmission of data.
- Comparison of random selection of network service providers with the preliminary simulation results of ENDA has proven that ENDA can make better offloading decisions, in terms of energy efficiency.

Another paper presents the design of cloudlet network and service architectures (IaaS) to study the performance impact of cloudlets in interactive mobile cloud applications, and it has further proven that the use of cloudlets improves performance gains in terms of data transfer delay and system throughput. Three different applications that are most commonly used in interactive mobile systems were used as part of the baseline in the simulation, i.e., file editing, video streaming, and collaborative chat (Fesehaye, Gao, & Nahrstedt, 2012).

Figure 3 shows the three types of network topologies in the context of interactive mobile cloud applications.



(a) (b) (c)

Figure 3. (a) Current mobile network topologies; (b) User-Cloud communication; (c) User-Cloudlet communication.

Two scenarios were considered, one was a static setup and the other was mobile. Three applications were put through both scenarios and tested over a cloudlet-based network and a cloud-based network. Overall, results have shown that when the nodes were static, systems that adopt the cloudlet scheme produced a lower request transfer delay and higher throughput, as compared to systems that adopt the cloud scheme. In addition, when the nodes were mobile, the cloudlet-based approach outperforms the cloud-based approach when the maximum number of cloudlet hops is 2. But when the maximum number of cloudlet hops exceeds 4, the cloudlet-based approach performs poorly for some of the requests made; although in general, the performance of cloudlet-based approach is still better (Fesehaye, Gao, & Nahrstedt, 2012).

THIS PAGE INTENTIONALLY LEFT BLANK

III. EXISTING NETWORKS AND PROPOSED MITIGATIONS

From our background research and literature review, we find that the scope of data used in naval operations is very extensive (C2, PNT, METOC, and others) and it would be a challenge to cover the entire scope. Therefore, for our research, we will focus on C2 data only. This will form the baseline for our study as well as our data model.

SATCOM provides the backbone platform for naval communications from the sea to the shore and vice versa. It is crucial for us to understand the technology and trends so that we can work with it.

In order to incorporate the cloud into our architecture, it is also important for us to understand cloud response time in the practical usage. With this understanding, we can make a better decision in meeting the response time we require from a cloud infrastructure.

A. APPLICATION OF C2

The following subsections provide more details on C2 and its data profile as a foundation to bridge over to its application.

1. C2 OODA Loop

In order to automate some of the C2 functions, data fusion (Waltz, 1986) of the various sources of information (i.e., merging and making sense of the incoming information) is essential to providing accurate and timely situation awareness to the commander. The order in which the information arrives into the system might paint a completely different picture to the commander. Besides other factors, the order of information arrival may be affected by delays in the communication link.

We review the high-level requirements for the Observe, Orient, Decide and Act (OODA) Loop (le Roux, 2008) as follows:

(1) Observe

- Sensor information and intelligence reports.
- Protocols for data and information exchange.
- Communication latencies.
- Data and information fusion, including multi-sensor, multi-target tracking.

(2) Orient

- Situation awareness.
- Display of data and information.
- Presentation of reports.

(3) Decide

- What-if analysis support for evaluating different options (effects analysis).
- Automated decision support systems - Agents assisting with relevant, timely, contextual and succinct information.
- Collateral damage assessments.

(4) Act

- Protocols for orders.
- Communications and latencies.
- Target-to-shooter association.

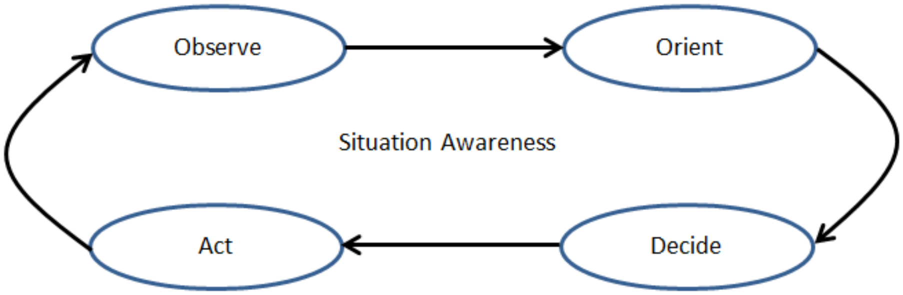


Figure 4. Situation Awareness in context of OODA loop.

Accurate situation awareness is essential to keep the loop going. If situation awareness is not able to be sustained (due to communication latencies in our context), we can see that the line between "Orient" and "Decide" would break. Without accurate information from the C2 system, it hinders the decision making of the commander, affecting all processes in the loop.

2. Data Profile and Characteristics

C2 heavily influences communication middleware requirements with its unique combination of needs. These include meeting real-time response to meet specified deadlines, fault tolerance to withstand damage and failures, scalability to accommodate dynamically increasing load as well as evolving requirements, and interoperability in order to manage software costs in the face of evolving hardware baselines (Swick, White, & Masters, 1999).

In general, C2 data can also be characterized into the following two categories (Swick, White, & Masters, 1999) as shown in Table 3.Table 3.

Table 3. Data categories.

Control Data	Streaming Data
a. Complex data order dependencies. b. Synchronous communication with great degree of reliability/delivery assurance. c. Data transmitted affects process/system state information necessary for fault tolerance. d. Sporadic data. e. Low periodic rate of transmission. f. Event oriented request/response information.	a. Limited data order dependencies. b. “Provider-receiver” data refresh nature. c. Data transmitted does not affect process/system state or is readily recoverable following process reconfiguration. d. Frequent data. e. High periodic rate of transmission. f. Periodic updates of an existing entity.

For this thesis, all C2 data and its usage profiles discussed are based on the following scenarios. Three data types are identified and further examples are given within each type. The examples are shown in Table 4.

Table 4. Examples of C2 data.

Text/Tracks	Images	Videos
• C2 messages • Tasking orders • Status updates • Readiness level • ISR reports • Location (Latitudes/ Longitudes/GPS coordinates) • Weather/METOC data • Inter-nodal/inter-agency information	• ISR Images • Sensor feeds • Map updates • Map overlays	• ISR Videos • UAV feeds • Sensor feeds

Besides understanding the characteristics and data profile of C2 data we also further explore how C2 data is being used in U.S. Navy operations. We study the QoS requirements of C2 data, and also the communications requirement that facilitates the transmission and receipt of C2 data between nodes of the system.

3. QoS Requirements

Developing QoS requirements of C2 applications (Wilson, 2007) begins with a study of the data exchange needs in an operational environment. The sensors within the C2 distributed system generate many different data types. These data types include intelligence reports, imagery, voice and video streams, commands and alerts, and resource management (e.g., health and status) messages. One of the critical challenges for data exchange for an effective tactical C2 capability will be reliably and efficiently routing the data over mobile, peer-to-peer, weakly connected networks in which links can be intermittent (e.g., hybrid network topology with multiple mobile ad hoc networks connected to SATCOM backbones). The purpose of the data exchange is twofold: (1) to achieve required levels of data consistency within geographically defined regions and/or within user-defined communities of interest in a given operational scenario, and (2) to share and manage distributed resources for joint service coordinated actions. The distributed resources include sensors that develop or collect data and the nodes that use the data, such as weapons and command and control systems.

For our discussion, we look into textual, imagery, and video data, which are the main inputs to an effective C2 situation picture. For textual reports, extensible Mark-up Language (XML) is commonly used as data representation (structured form) and messaging between entities/systems. Computers are able to interpret and process structured rather than unstructured data. Programming languages such as C++ and Java provide application programming interfaces (APIs) that support the construction and parsing of XML messages. This is especially the case if you have a schema where information can be represented.

4. Communication Requirements

There are significant differences in terms of the requirements, the number of users, the environment, equipment and support, etc., when it comes to conducting telecommunication at sea compared with on land. The land-based facilities are usually equipped with high-performance wired connectivity while sea-based platforms solely rely on wireless connectivity. However, land-based facilities may be utilized as wireless base stations or relay facilities for coverage to their immediate surroundings.

Some of the identified major technological objectives to meet high communication requirements are:

- Extension of the coverage and range at sea for terrestrial wireless systems/ technologies.
- Proposal of suitable SATCOM alternatives to complement terrestrial wireless systems/technologies.
- Seamless handover and roaming in both intra- and inter-systems.

Some of the currently available digital maritime telecommunication technologies that are used commercially are covered in Table 5.

Table 5. Currently available commercial communication techniques (from Bekkadal, 2010).

System	Communication form	Data rate
NAVTEX	HF, MF	300 bps
DSC	VHF	1.2 kbps
GPS	Access via NMEA 0183	4.8 kbps
AIS	VHF	2 x 9.6 kbps
EPIRB	Short messages (Satellite)	100 bits / hour
SSAS	Short messages (Satellite)	100 bits / day
SafetyNET	NAVTEX over Inmarsat	100 messages / day
Some ships have digital data links via Satellite (Inmarsat, VSAT…)		

B. SATELLITE COMMUNICATIONS

This section provides the background of SATCOM which is essential for naval communications. This is to build on the understanding of SATCOM which is helpful in our work.

1. Commercial Satellites as Supplement to Military SATCOM

It is not cost effective and efficient to put more communication satellites into orbit just to increase bandwidth. In fact, we would need to adopt an approach which would allow us to boost data rates as well as to allow satellites to handle more data channels. One of the most effective techniques is to use new communication bands, e.g., the Ka-band, which leverages on its large bandwidth capability to be able to send more data in lesser time.

Ka-band technology can transfer data as quickly as 50 Megabits per second (Mbps), as compared to the 492 Kilobits per second (Kbps) of the L-band and Ku-band satellites (Costlow, 2011). Furthermore, L-bands and Ku-bands are almost 90 percent saturated, preventing more signals from being able to use the pipe. Despite being able to provide data rates that are 100 times better, Ka-band does have its disadvantages. For example, it is more susceptible to environmental attenuation. Also, other techniques have to be implemented to complement Ka-band links in order to achieve the optimal performance.

One of the methods is to implement adaptive coding modulation, which provides feedback and increases or decreases the modulation rate to improve the burn through rate. Configuring TDMA to work in deterministic mode enables a large number of users to fully utilize the satellite bandwidth. Another alternative to improving efficiency could be data prioritization. Critical data which supports mission requirements are transmitted first, followed by non-critical data. Also, knowing when to fall back to using L-band or Ku-band is beneficial to data transmission during adverse conditions.

2. INMARSAT

INMARSAT is a geostationary system that has four operational satellites. The satellites are all close to the equator and have overlapping regions of coverage around the globe (FAO Corporate Document Repository, 2001). Its coverage ranges from the Pacific and Indian Oceans to the Atlantic Ocean, providing an almost universal coverage.

INMARSAT offers a number of different types of service formats using the same satellites (FAO Corporate Document Repository, 2001):

1. Inmarsat A (analogue) or Inmarsat B (digital) is widely used by large vessels. Service formats include voice, facsimile, and high-speed data transmission in both send and receive modes. Inmarsat A or B provides an "end to end," or duplex type of service, such that the sender and receiver are able to communicate in an almost immediate real-time contact.

2. Inmarsat M provides similar services to A and B, but of a smaller and lower speed format.

3. Inmarsat C is very different from the other formats offered. Instead of an "end to end" system, it is more of a "store and forward" system, where messages are stored in intermediate locations before they get forwarded to the final destination. Transmission time is typically about 5 minutes, which is not suitable for voice communications. This solution, on the other hand, is suitable and less expensive for Emails and messages. Inmarsat C also offers message reporting mode and data reporting mode. The former allows sending of free-format messages while the latter allows for transmission of 16-bit packets of data.

C. CLOUD RESPONSE TIME

For real-time and highly interactive applications, fast response time is a key requirement for satisfactory performance. Response time (Li, Yang, Kandula, & Zhang, 2010) is the total time the application takes from when a user makes a request until he receives a response. In the following subsections, we illustrate some of the challenges in meeting response time requirements and also discuss the response times of services provided by commercial providers. Cloud service providers also introduce new architectures to improve the response time.

1. Challenges of Meeting Response Time Requirements

In recent research (Dey, Liu, Wang, & Lu, 2013), Cloud Mobile Gaming (CMG) and Cloud Mobile Desktop (CMD) applications are used to investigate the viability of using a public cloud server provider, Amazon Web Services (AWS). The results collected give us some idea of the challenges of meeting the response time requirements of (especially real-time) applications.

By taking the average of the users' opinion, the response time requirements are listed in Table 6.

Table 6. Response Time Requirement for CMG and CMD.

	CMG	CMD	
		Slide Show	Typing
Acceptable	440 ms	835 ms	390 ms
Excellent	280 ms	445 ms	125 ms

They have conducted various experiments to find out the response time of the cloud connected to a 3G network and WiFi hotspot provided by a mobile network operator.

For CMG, the streaming of video using the commercial video conferencing software Skype was used. High frame rate is important for good gaming experience, which can be emulated using conference video. In Figure 5, we can see that the average response times for both 3G and WiFi is higher than the acceptable range, which means that it would not be a satisfactory user experience.

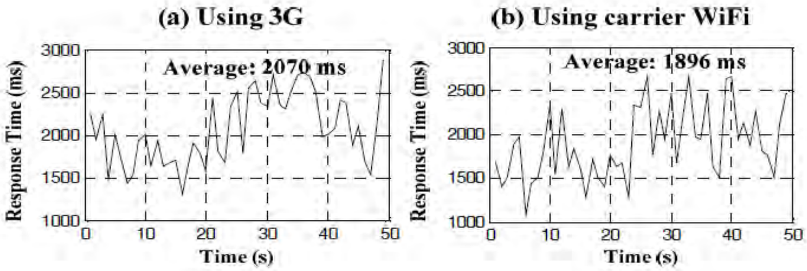


Figure 5. Response time using Skype to stream CMG video.

The remote desktop application Citrix is used for the experiments conducted for CMD. In Figure 6, we can see that the user viewing a slide show would need to wait for an unacceptable time before there is a response to his instructions to the desktop. He would need to wait for a while after executing his instruction before the next slide appears. For the typing experience, it is also unacceptable for the user to wait for half a second before his input to the keyboard appears on the screen.

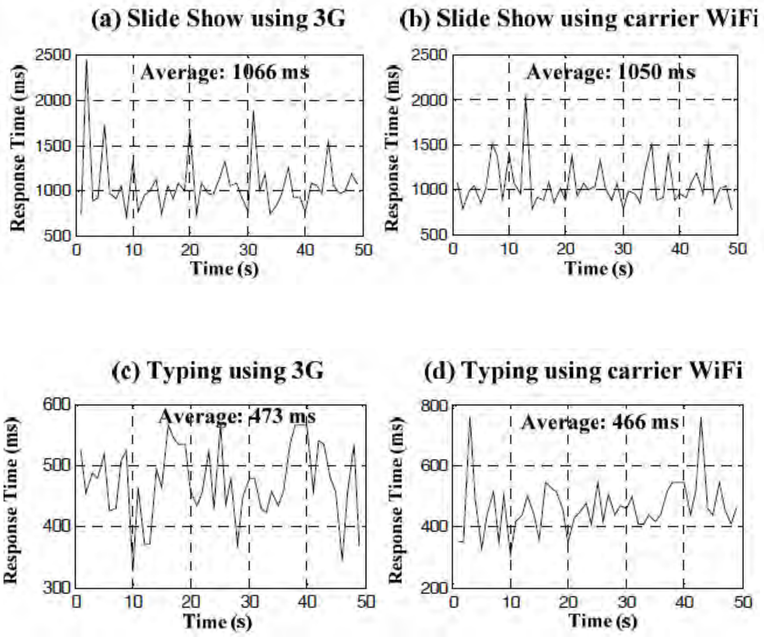


Figure 6. Response time using Citrix as CMD application.

From the results of the experiments, we can appreciate the challenges to achieving the desired response time we want, specific to each kind of application.

2. Response Times of Amazon EC2 Instances

In a 2010 study (Alhamad, Dillon, Wu, & Chang), a series of experiments on Amazon Elastic Compute Cloud (EC2) are reported. The response time of five types of Amazon EC2 instances was evaluated. In the paper, CPU performance is used as the main parameter for cloud performance, and the execution time of a Java application over five types of Amazon EC2 instances was recorded. The response time was recorded every two hours during several days of experimentation.

Different types of virtual machines in terms of CPU capacity, RAM size, and disk size were used. The attributes of virtual machines that were used in the experiments are listed in Table 7.

Table 7. Attributes of Amazon EC2 instances.

Instance Type	EC unit	Cores	Architecture	Disk (GB)	RAM (GB)
Small	1	1	32 bits	160	1.7
Medium (H-CPU)	5	2	32 bits	350	1.7
Large	4	2	64 bits	850	7.5
Extra Large	8	4	64 bits	1690	15
Extra Large (H-CPU)	20	8	64 bits	1690	7

It might be intuitive that the response time from the Extra Large (H-CPU) instance would be the best. In Figure 7, the results of Small, Large, and Extra Large (H-CPU) instances are shown. It is also interesting to point out that we can get better stability of the response time with better CPU resources.

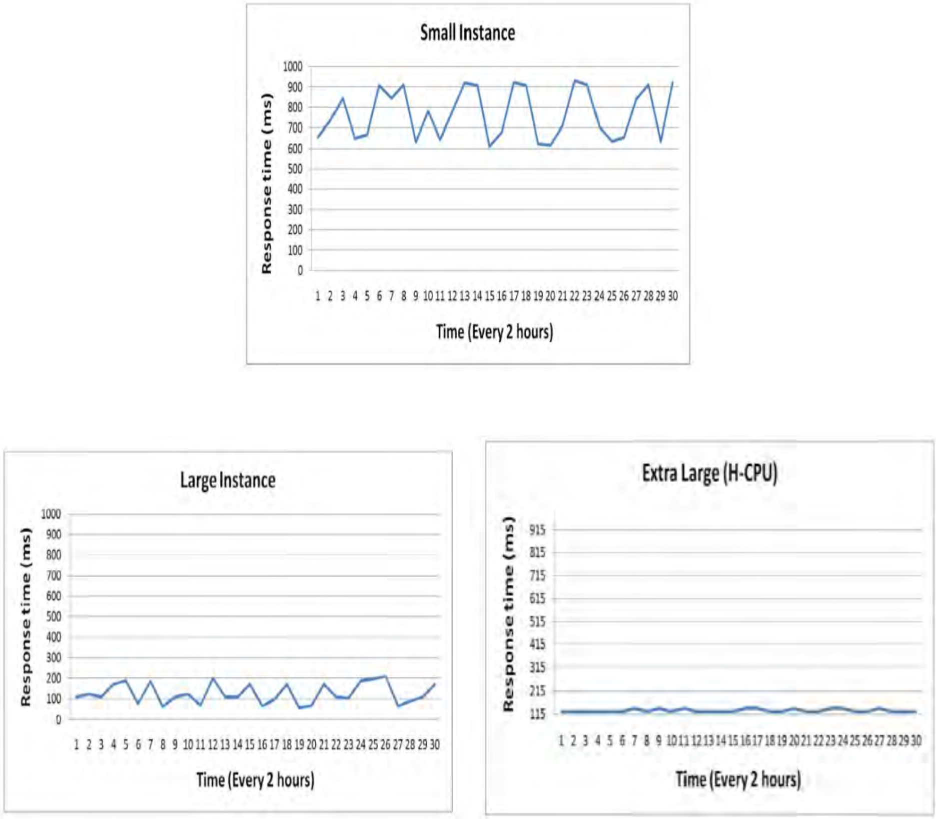


Figure 7. Response time of Small, Large, and Extra Large (H-CPU) instances.

3. Speed Testing on Amazon EC2 and S3 across Regions

Another study (Chen, 2013) investigates the upload/download speed between different AWS regions to see the differences in speed when transmitting data between EC2 and Simple Storage Service (S3) buckets in different regions. In Table 8, we can see the average results for the 10 Megabyte (MB) upload test (in seconds).

The X-axis data is the S3 bucket region and the Y-axis data is the EC2 region. The results show the penalty physical distance has on performance. As expected, the best upload time (highlighted in red) occurred when both the EC2 instance and S3 bucket were located in the same region.

Table 8. Ten MB upload test across different regions.

	Virginia	California	Oregon	Ireland	Singapore	Japan	Australia	Brazil
Virginia	2.628	11.743	4.02	5.068	15.947	11.204	28.446	10.141
California	4.611	1.061	1.142	9.409	13.024	5.696	7.121	8.831
Oregon	4.376	3.263	0.684	5.429	11.82	5.938	11.954	9.891
Ireland	5.291	18.377	8.416	0.755	17.025	12.852	14.2	12.769
Singapore	14.62	26.244	9.345	40.317	0.851	4.397	6.007	19.763
Japan	9.918	9.629	7.078	11.909	7.711	1.051	9.263	14.575
Australia	12.528	17.989	9.973	182.011	183.682	40.717	0.812	216.164
Brazil	7.482	28.133	9.581	15.415	23.819	14.043	10.396	0.789

The results from this study help to reinforce our belief that physical location to our data sources is very important to improve the communications.

4. Using Amazon CloudFront To Improve Response Time

Amazon provides the option for businesses and developers who want to distribute content to end users with low latency and high data transfer speeds. If you are willing to pay more, you can get the mentioned benefits from Amazon CloudFront.

Amazon CloudFront is a content distribution network (CDN) which is tightly integrated with Amazon S3. It is designed specifically to improve static content delivery. S3 is designed to easily store and retrieve data. When S3 is used together with CloudFront, S3 becomes the offsite backup of CloudFront.

CloudFront moves the S3 content to the network “edge,” geographically closer to the end user, which helps reduce latency as shown in Figure 8. It is a pull model where content is pulled from S3 to the edge upon first request and it expires in 24 hours by default.

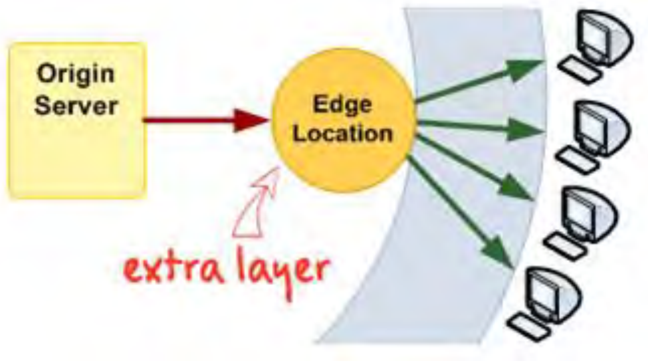


Figure 8. Illustration of an edge cache nearer to end users.

D. SCENARIOS ANALYSIS

Mission threats help to identify system functionality and a reasonable operating environment in which the system is expected to perform. Mission threats should also facilitate in generating requirements for C2. We have identified representative operational scenarios to demonstrate the capabilities of a C2 architecture utilized to support the mission. Taking into consideration the different usages of C2 data, we look at two different scenarios, one describing a peacetime surveillance operation while the other depicts a mission-oriented operation.

1. Peacetime Use Case

Security, stability, and safety have been, and continue to be, the objectives of the U.S. Navy's maritime activities. The identified key functions and missions are:

- **Maritime Security**. Operations such as maritime patrols, interception operations, and support of U.S. Coast Guard operations.

- **Sea Control.** The U.S. Navy is forward deployed and protects maritime access to vital sea lanes and operating areas.

- **Projection of Power.** The U.S. Navy is prepared to ensure that national interests are protected. The navy aims to be versatile and flexible to rapidly and effectively deploy and sustain forces in and from multiple dispersed locations in order to respond to crisis, contribute to deterrence, and to enhance regional stability.

- **Search and Rescue (SAR).** Due to the challenges in SAR missions caused by extreme distances, limited infrastructure, and assets, the U.S.

Navy provides support to such missions conducted and led by the U.S. Coast Guard, and as directed, in support of international partners.

- **Humanitarian Assistance/Disaster Response (HA/DR).** With the increased risk in the occurrence of a maritime or environmental disaster, the U.S. Navy has to be prepared to support these critical missions and they must be trained and sufficiently equipped to lead, facilitate, or provide assistance to other agencies. The U.S. Navy remains ready to support critical and likely missions, such as pollution response and SAR; integrated planning efforts with local, state, federal, and native communities; strengthen interoperability with the U.S. Coast Guard and international partners; and develop processes, procedures, joint training, and exercises to gain operational proficiency (Operations, 2014).

A possible realistic scenario, with the U.S. Navy projecting its capabilities and expanding its presence, is described here.

An action group consists of a guided missile destroyer (DDG) and a few missile frigates (FFG) patrolling the area of operations (AO). The DDG could have an aerial platform to support over-the-horizon search and targeting capabilities. The purpose of this operation is to protect the friendly sea line of communication by conducting maritime patrols in the AO. At the same time, each platform achieves the status of being ready, to prepare for any unforeseen hostile situations.

The operational view as shown in Figure 9 is a high-level representation of how the peacetime operation is conducted, depicting the interfaces between the various naval platforms involved, and also the associated C2 activities. The functionality for Intelligence, Surveillance, and Reconnaissance (ISR) is also discussed.

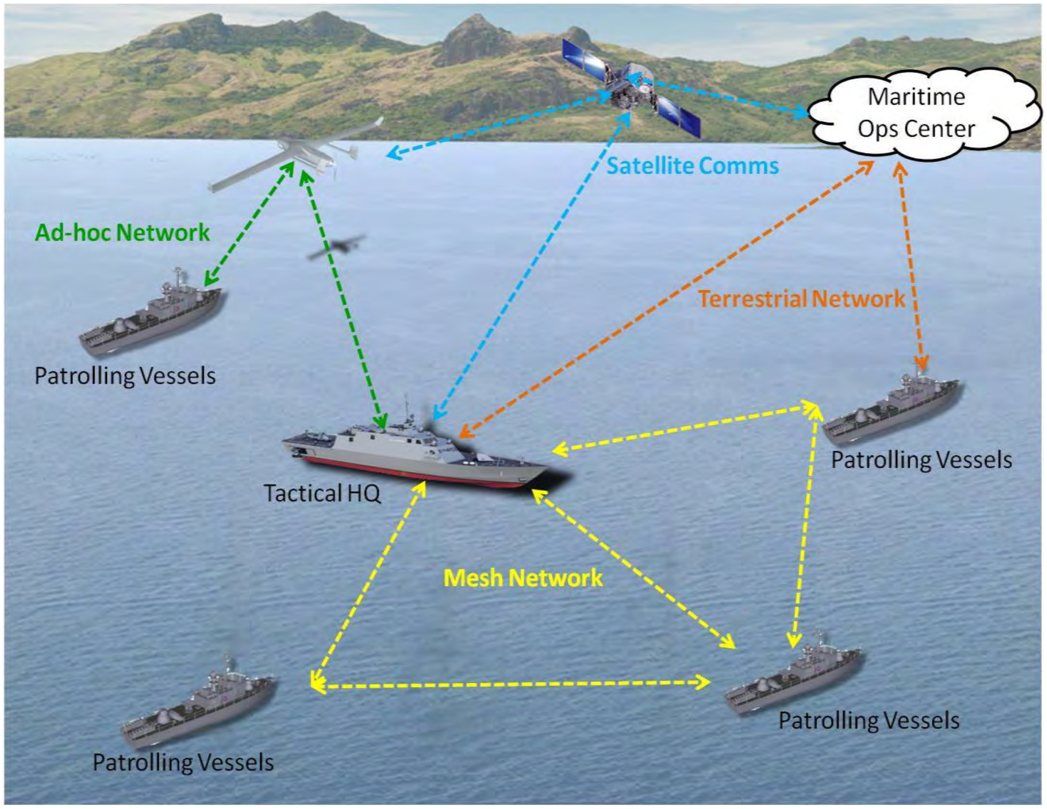


Figure 9. Operational view: Maritime patrol operation.

This scenario assumes that the Tactical Headquarters (Tact HQ) is residing on board the DDG. This Tact HQ will receive higher order instructions from the Ops Center on shore and it is also responsible to update status reports and other C2 information back to the Ops Center. The primary mode of communications used would be via SATCOM links, and C2 data would be uploaded from/downloaded to a main server.

The other frigates act as patrolling vessels, and for downstream orders and/or upstream status updates, they communicate back to the Tact HQ. The primary mode of communications would be Line-of-Sight (LOS) communications.

In typical cases where there are no ad-hoc situations that require the use of ad-hoc networks, patrolling vessels receive tasking orders and instructions from the Tact HQ and conduct necessary patrols in their respective AOs. Surveillance results and intelligence information, in the form of still images as well as videos could be consolidated and sent to Tact HQ for record and analysis. Additionally, location updates of respective patrolling vessels are also reported back to Tact HQ for updates to situational awareness. When there are any updates in tasking orders, C2 information is distributed to all patrolling vessels in the form of messages. At the same time, Tact HQ is also required to periodically update the Ops Center on its current status, and synchronize its database with the Ops Center.

In cases where during patrol, ISR assets on a patrolling vessel identify an unknown target, the unit will have to notify Tact HQ about the target. Additional information about the target, such as the size, the type of vessel, etc., can be sent to Tact HQ in the form of still images and videos provided by the sensor units. This will allow the Tact HQ to conduct identification of friend or foe and formulate Rules of Engagement accordingly. At the end of any situation, all patrolling vessels and Tact HQ will assess their readiness level and provide a report back to the Ops Center.

A summary of the flow of data is represented in Figure 10.

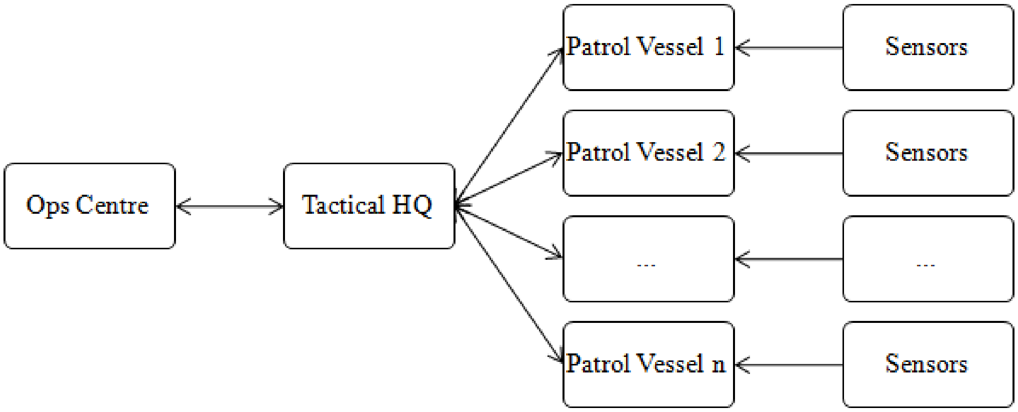


Figure 10. Brief summary of data flow.

2. Piracy Interdiction Use Case

An Expanded Maritime Interdiction Operations (EMIO) scenario (Perkins, 2013) is used for our study to illustrate the importance of C2 capability in the operation. The scenario highlights Maritime Tactical Command and Control (MTC2) interaction between a Maritime Operations Center (MOC) ashore and two Carrier Task Force (CTF) groups afloat. The scenario starts with reliable intelligence reports to indicate that piracy activity is likely in the Area of Responsibility. One example of such activity is when a commercial ship carrying hazardous cargo has been hijacked but no further information of the identity of the vessel is available. As a result, the Commander has issued an order that all commercial ships carrying hazardous cargo are Critical Contacts of Interest (CCIs).

Use cases of MTC2 capabilities that are employed during this EMIO mission include:

- Visualization, updating, and distribution of CCI (new track) and EMIO mission information.
- Discovery and identification of tracks and addition to the COP.
- Data exchange between the data sources and the data centers on the MOC/platforms.
- Data fusion of reports to maintain unique tracks.
- Live feed from UAV can augment the track information with live images and video.
- Dissemination of new EMIO mission requirements in the COP.
- Force readiness assessment and allocation of resources for EMIO platform-boarding operation.

This scenario shows that MOC ashore is interacting with the two Carrier Task Forces, CTF1 and CTF2 afloat. The UAV is being employed by CTF1 to extend its sensor range. In the scenario, the UAV is launched from and under the command CTF1. The commercial vessel is detected by the UAV and the information is transmitted back to the CTF1/DDG. The information (new track) is entered into the C2 system. This information should be relayed to the MOC through the SATCOM.

MOC site hosts a central organizing structure, which we can call the “Data Hub,” where data from different sources is organized and fused to form the common operating picture. All other platforms should be synchronized with the “data hub.”

The scenario follows with the assignment of mission to nearest CTF2/DDG with the readiness level sufficient to board ship. However, there is an unanticipated SATCOM disconnection of CTF2/DDG, the new track is not updated in its operation picture. The DDG cannot respond to the mission promptly, and no one knows when the communication would resume.

Another consequence which might seem to be less severe is the loss of the live image or video feed of the suspicious track. In today's C2 domain, this could be the less prioritized information to get through. However, it could be a great value-added function, as most people would agree that "a picture is worth a thousand words."

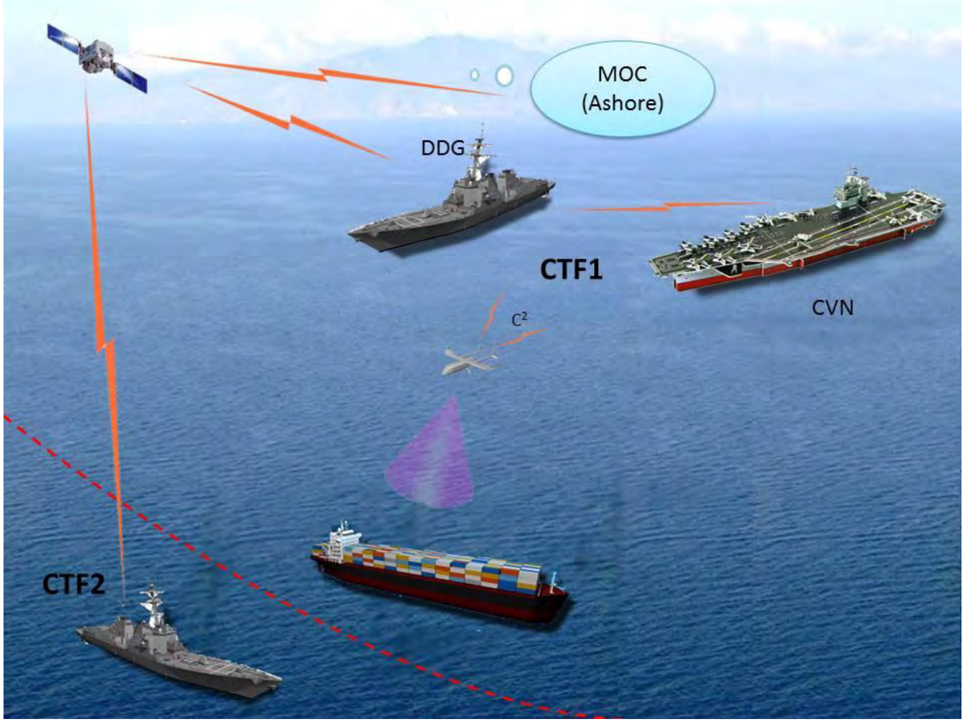


Figure 11. Operational view: Piracy interdiction operation.

E. PROPOSED MITIGATIONS

To be able to support real-time and near real-time data usage, we want to leverage on the benefits of caching. With caches, data that has been previously requested can be stored locally and the next time this data is requested again, it will be more readily available. Therefore, one of our proposed strategies is the implementation of caches.

In many cases, it is not feasible to create a cloud infrastructure that is within the range of every node. As mentioned in the related works of Chapter II, we observe that cloudlets seemed to be analogous to the CloudFront concept described in the previous subsection of this chapter. Therefore, we will want to deploy cloudlets to be within close proximity of every node, to close the gap by extending each node's maximum connectivity range. This way we can leverage on their connectivity to the remote cloud server. In addition, cloudlets can be incorporated with caches to supplement the nodes connecting to cloudlets to access information, and thereby support real-time and near-real-time cases. On the other hand, if data is not available in the cache of the cloudlet node, the cloudlet node can request the information from the remote cloud server on behalf of the nodes via the satellite.

With the aforementioned considerations, we propose to implement caching and cloudlets as part of the mitigation strategies. These are further discussed in the following subsection.

1. Local Caches

Caches are deployed on each node to facilitate the requests made by each node to the remote cloud server. When a node requests certain data, it will first look at its own cache. If the data is not available on the local cache, it will make a request the required data from the cloud server. Once this data is retrieved, it will be stored in the local cache of the requesting node. The next time the same data is requested, it will be available in the local cache and this shortens the overall response time of generating a request and receiving a reply from the server, thus improving the performance of the data transmission. Figure 12 briefly depicts this mitigation.

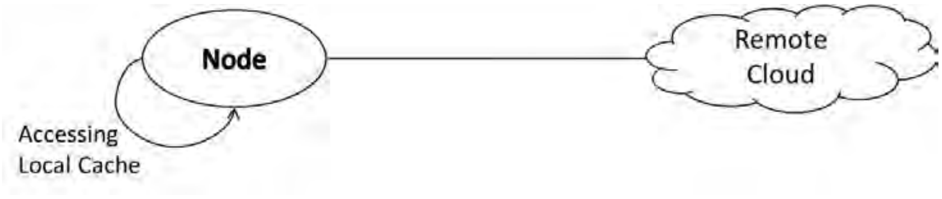


Figure 12. Deploying local cache on platform.

2. Deploying Cloudlet in the Architecture

The cloudlet is assumed to be deployed on the Tact HQ (referred to as the cloudlet node) of the action group, although any node can also take on the role of a cloudlet. This cloudlet node is responsible for the communication back to the remote cloud server for access of data. The other naval platforms (commonly referred to as nodes) can connect to the cloudlet node to access information that they require. This mitigation takes into consideration two factors, the number of connections to the cloudlet node and the data requests by the nodes to the cloudlet node, then to the remote cloud. The aim is to minimize the connections back the remote cloud, reducing over-utilization of the bandwidth. Figure 13 briefly depicts this mitigation.

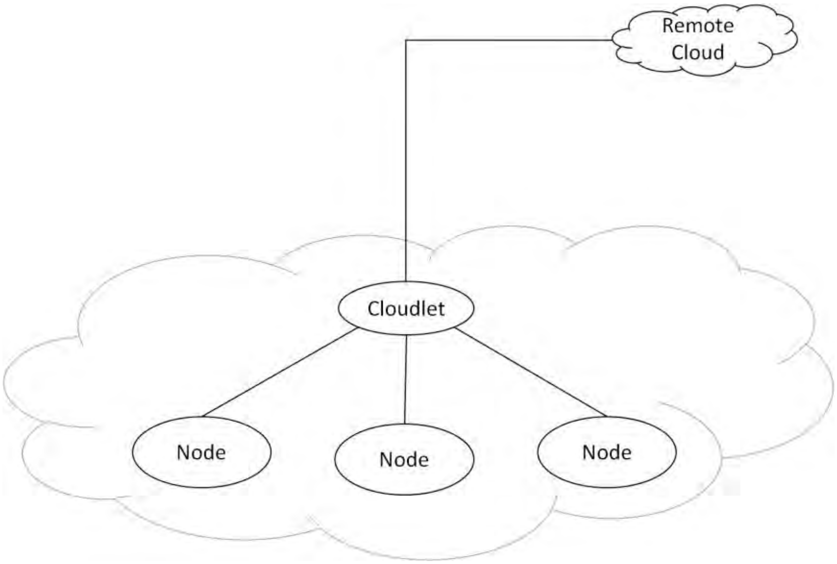


Figure 13. One node acting as a cloudlet.

These strategies are proposed based on our preliminary research and also the available methodologies in the current mobile-cloud architecture. The benefits and advantages suggest that these strategies are able to contribute to meeting our thesis goals. But in order to verify our claims, we will need to conduct a deeper feasibility study of the strategies and this process is described in the next chapter.

IV. MODELING AND SIMULATION OF NAVAL COMMUNICATIONS AND CLOUD ARCHITECTURE USING QUALNET

Modeling and simulation can assist us to get information about the expected behavior of our proposed architecture without building the real-life physical system. Since the latest naval communications involve SATCOM and incorporate cloud architecture, our modeling and simulation software has to contain models that can simulate these elements.

Every modeling and simulation software package has its strengths and weaknesses. For example, open source OMNeT++ offers an Eclipse-based IDE (integrated development environment) which is programming friendly. Therefore, the existing models can be customized and new models built (OMNeT++, 2013). The downside is that the existing models are lean, and therefore, much effort has to be put into programming and building new models which we need but are not available. On the other hand, commercial products like OPNET and QualNet provide extensive libraries of models which make them more attractive. Although they are customizable, their support for programming is less extensive. Given our constraints, modifying existing models or protocol models is more achievable than adding new models into the software. In addition, the user community online forum does not extensively discuss adding new models: even the official forum does not seem to readily support it.

A. SOFTWARE AND HARDWARE

In the following subsections, we describe the software and hardware we selected for our modeling and simulation.

1. Software Used

We used QualNet 7.1 with an educational license for our modeling and simulation. QualNet is a communications simulation platform which allows us to create a virtual network environment or scenario that we envision.

QualNet provides the libraries of models such as Application models, Protocol models, Satellite models and Wireless models, which can be useful for our simulation. It is a discrete-event simulator. This means that the operation of a system is modeled as it proceeds over time by a representation in which the system's state changes instantaneously when an event occurs. An event is defined as an instantaneous occurrence that causes the system to change its state or to perform a specific action (Scalable Network Technologies, 2013). Fundamentally, the class used to represent an event is called "Message." In QualNet, Packet events are used to simulate the transmission of packets over the network. A Packet event is defined using the "Message" class.

QualNet also provides a basic statistical graphing tool which displays the metrics collected in a simulation. This visual tool is useful for our analysis of the collected results.

Following are some desirable features that cannot be achieved using QualNet. Therefore, workarounds are required to simulate the operations we want.

- Cache within a platform which can be used to simulate a cloudlet.
- Data centers that can be used to simulate a cloud.

Microsoft Office Excel spreadsheet is also a popular and useful tool to compute the data required, especially those calculations which are huge but repetitive.

2. Hardware Used

The machine used to run the QualNet model is an Intel Core i7-4500U Dual Core Dell laptop with 16GB of RAM (random-access memory), running on a 64-bit Windows 8 Operating System. The full QualNet installation requires around 900 Megabytes (MB) of hard disk space.

B. APPROACH AND ASSUMPTIONS

The objective of our model is to test whether the implementation of caches will benefit in a DIL environment. Intuitively, the volume of data, the bandwidth of the communication link, and the response time of our data traffic (from source to destination and back) are directly related to one another. For example, with a fixed amount of bandwidth, the higher the data volume, the longer it will take for the source node to receive a reply from its destination. Similarly, the lower the data volume, the faster will be the response time.

Cache performance (Hennessy & Patterson, 2011) is generally measured using average memory access time (AMAT) as follows:

$$AMAT = hit\ time\ +\ miss\ rate\ \times\ miss\ penalty \tag{1}$$

In order to fit our requirement, we need to relate this formula with the consideration of the volume of data, the bandwidth of the communication link, and the response time of our data. The following paragraphs step through the process of deriving a formula that measures the response time for our model.

In web caching, there are generally three kinds of data, static, semi- static, and dynamic. They are categorized based on life time of the data or Time-To-Live (TTL).

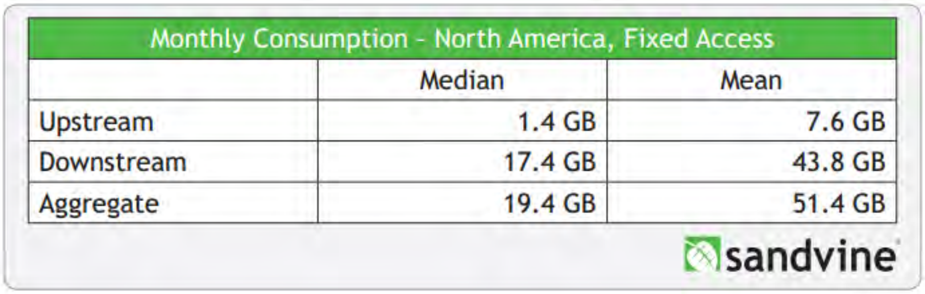
- Static: the data does not change in its life time (TTL = infinity). For example, static web page with no dynamic contents. The data does not change for every request, thus, caching is most useful for this kind of data.

- Semi-static: the data does change but not that often (0 < TTL < infinity). For example, weather forecast webpage which is updated every two hours. The data does change for some requests; thus, caching is still useful but not much as static data.

- Dynamic: the data does change for every request (TTL = 0). For example, real-time stock price webpage which presents different information every second or less. Caching dynamic data is the least useful.

We want to model our operating environment as close to the current environment as possible, but due to the security classification of the existing data set, we were only able to take reference from public sources. Our data usage profile will be based on actual

Internet traffic, with reference from Sandvine (2014), a broadband equipment company. This will give us an approximate representation of the existing data usage profile of the U.S. Navy. However, by plugging in accurate navy requirements, we can get navy-specific results. Sandvine's bi-annual report measures the average Internet traffic demand of a general Internet user for the first half of 2014, and it also provides a categorical breakdown of the traffic demand as shown in Figure 14.



Monthly Consumption - North America, Fixed Access		
	Median	Mean
Upstream	1.4 GB	7.6 GB
Downstream	17.4 GB	43.8 GB
Aggregate	19.4 GB	51.4 GB

Figure 14. Monthly Consumption Figures (per individual user) – North America, Fixed Access (from Sandvine, 2014).

We reckon that upstream consumption is probably the data demand for uploading. Since upstream consumption is small as compared to downstream, it is reasonable to use aggregated data consumption as our total data volume. From Figure 14, we have the average monthly consumption of data at 51.4GB. Hence, the total data demand rate for our test set is calculated to be approximately 21,300 bytes per second (this number is a simple conversion of the data demand from month to seconds).

In Figure 15, we see that real-time traffic, covering applications which require "on-demand" data, takes up about 59 percent of the total demand. Communications traffic, consisting of real-time chat, voice, and video communications, takes up 13 percent of the total traffic demand. These two categories of data fall under the data type of dynamic data, as their content are continuously updated, making them non-cacheable. The rest of the categories will be broadly categorized into cacheable type of data, taking up 28 percent of the total data demand. This assumption closely aligns to the findings from Wessels (2001) where between 35 and 70 percent of all requested objects are cacheable. To understand how the ratio of cacheable objects affects the response time in our experiment, we intend to run our model over the range of 30 to 70 percent of cacheable data. This will give us a good coverage of data with the characteristics of being cacheable and non-cacheable.

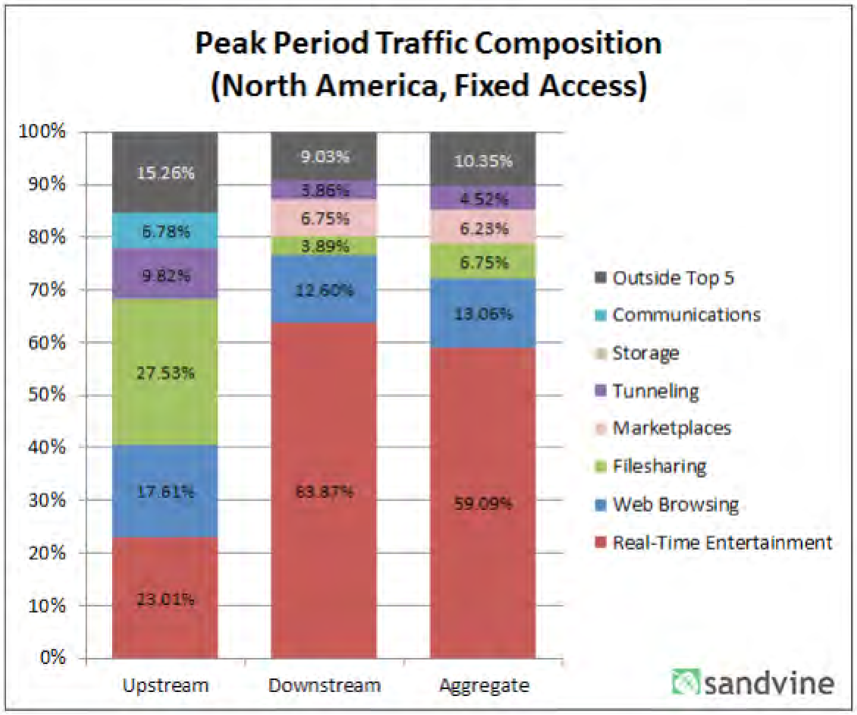


Figure 15. Peak Period Aggregate Traffic Consumption – North America, Fixed Access (from Sandvine, 2014).

Web caching (Beaumont, 2000) can provide significant benefits to both the end user and the service provider. The end user can enjoy a faster surfing of the web if the requested objects are in the cache. For the service provider, there will be savings in the bandwidth. The mentioned benefits can be achieved only when the requested objects from the web are available in the cache. This is the probability of the requested objects being found in the cache, which is called probability of a hit or hit ratio, P(hit). The hit ratio is dependent on several factors, such cache size, number of objects available in the Internet, average size of object, and percentage of cacheable objects, etc.

Ideally, P(hit) should take into consideration the stochastic behavior of each TTL value, which is renewed every time a new copy of data is downloaded from the remote server. To achieve this, we would need to run stochastic simulations for caching, taking into consideration inputs, such as cacheable data volume, probability of data being cached, probability of data being accessed, and different TTL values to test for static and semi-static data. However, this is not the approach we are taking.

Our assumption is that the TTL is much greater than the inter-access time. This means that only the most-recently accessed items will be in the cache and the only reason to retrieve a data from the remote cloud is because the needed object is not in the cache. Therefore, we would divide the data type into two categories, cacheable and non-cacheable. The P(hit) values from the Zipf distribution (Beaumont, 2000) shown in Figure 16 would be used in our simulation calculations.

Zipf distribution is a popularity model, where the probability of an object being requested is proportional to the rank of that object. Figure 16 uses this Zipf popularity model, where hit ratios are plotted as a function of cache size for 5 KB objects with four Alpha values (a higher Alpha value means that popular objects are much more popular). This graph does not take into account the expiration times of objects. It provides us reasonable hit ratios that we require to run our simulation.

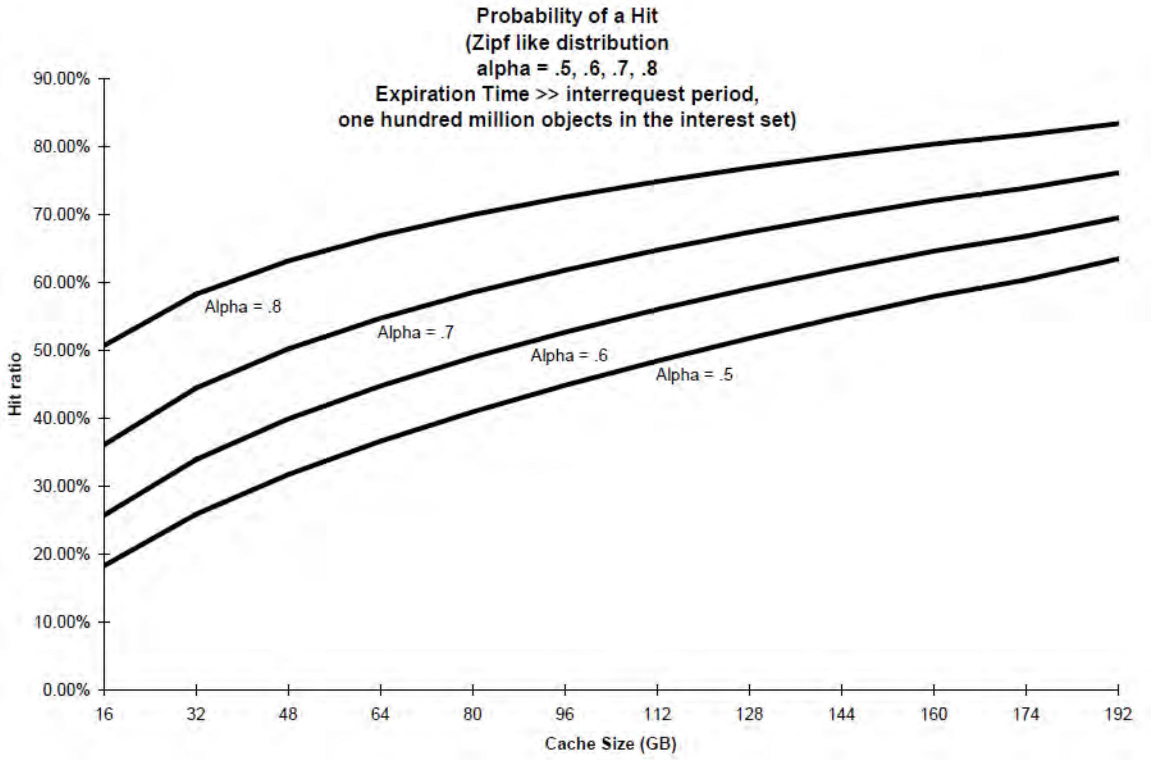


Figure 16. Hit ratio against cache size (from Beaumont, 2000).

The corresponding data volume of each category is determined simply by the following formulas:

$$\eta = \%Cacheable \times Total\_data\_volume \tag{2}$$

$$\mu = (1 - \%Cacheable) \times Total\_data\_volume \tag{3}$$

where η is the Cacheable data volume and μ is the non-Cacheable data volume.

We define the total data requirement, TR to be the data demand that will go through the SATCOM link, such that

$$TR = \mu + \eta \times (1 - P(\text{hit})) \tag{4}$$

where η is the Cacheable data volume and μ is the non-Cacheable data volume.

With values of Cacheability, P(hit) and TR set up, we can get the average response time, ζ using the following formula.

$$\zeta = \delta \times P(\text{hit}) + (\delta + \psi) \times P(\text{miss}) \tag{5}$$

where δ is the average local time, ψ is the average remote time, and P(miss) is the miss ratio or 1-P(hit).

The average remote time, ψ can be obtained via QualNet by plugging the total data demand rate, TR, as input to the model (via SATCOM). The average remote access time is then defined as the time taken for the source node to receive a reply after the request is made. The ψ values will be obtained with respect to the function of TR and Bandwidth (BW):

$$\psi = f(TR, BW) \tag{6}$$

With that, average response time, ζ becomes:

$$\zeta = \delta \times P(\text{hit}) + ((\delta + f(\text{TR}, \text{BW})) \times (1 - P(\text{hit})) \tag{7}$$

The typical response time for accessing local cache, δ is between 30 to 35 milliseconds according to an article from ScaleOut (2007). Our experiment uses a fixed value of 30 milliseconds as the average local access time.

C. MODEL DESCRIPTION

Based on the modeling approach discussed in the previous section, we adopted an incremental approach, and prepared three test cases. The base case forms the baseline of the simulation; Case 1 includes the implementation of cache, while Case 2 includes the implementation of cloudlet. Each test case is discussed in detail in the following sub-sections.

1. Base Case without Cache

The base case models the scenario where caches are not implemented. The results from this base case form the baseline for our analysis in Chapter V. Since caches are not implemented, there is no cacheable data per se and the values for %cacheable and %non-cacheable are 0 percent and 100 percent, respectively. For the same reason, the hit ratio, P(hit) is zero. Therefore, the TR for the base case will be the total data volume that is going to the remote cloud server via the SATCOM. With that, the generic formulas for TR and average response time, ζ, presented in the previous section are reduced to:

$$TR = Total\_data\_volume \tag{8}$$

$$\zeta = \delta + f(\mathrm{TR}, \mathrm{BW}) \tag{9}$$

Basically, we are using our QualNet model shown in Figure 17 to measure the time taken for the remote server to reply after the source node initiates the request.

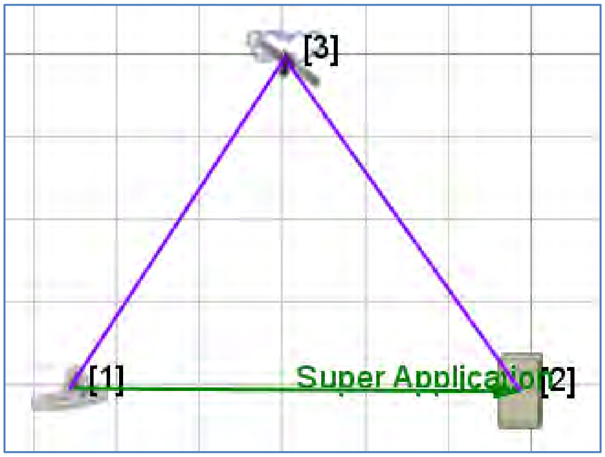


Figure 17. QualNet Model for base case without cache.

In our simulation, we vary the total data requirement, starting from ten users to 60 users and increasing in steps of ten users. We aim to find how the average response time will change with the increasing data requirement. Another parameter that we are varying is the available bandwidth of the SATCOM. By determining how the bandwidth affects the average response time will give us a good estimation of the minimum bandwidth requirement that we need for the given amount of data volume. This also gives us some idea of how the implementation of cache can overcome the effect on performance when operating under a limited bandwidth.

2. Case 1: Modeling with Local Cache

In this model, we assume that the naval platform has a local cache which would store some of the data objects. If the platform is requesting objects which are available in the local cache, the response time would be relatively faster than requesting from a remote server or cloud. But since we are not expecting everything in the Internet to be available in the local cache, the average response time of the requests would potentially be slower.

The QualNet model (from Figure 17) is modified to include a local cache at the node. The configuration of the node and the calculation of the response time are achieved using Excel. With the result received from the base case, we decide to keep bandwidth, BW, fixed at an optimal value of 2 Mbps, as the behavior with varying bandwidth is intuitive. Without considering the bandwidth, Equation (7) is potentially reduced to:

$$\zeta = \delta \times P(\text{hit}) + (\delta + f(\text{TR}) \times (1 - P(\text{hit})) \qquad (10)$$

The average response time is very much dependent on the TR and P(hit). Similarly, TR is directly proportional to the number of users using the network simultaneously, and we vary the TR from ten to 60 users, in steps of ten users, the same way as the base case. In addition to that, we vary %cacheable and P(hit). For %cacheable, we vary from 30 percent to 70 percent in steps of 10 percent. As for P(hit), we vary from 10 percent to 80 percent in step of 10 percent so that we can substantially cover the whole range of P(hit) for different Alpha values shown in Figure 16.

Through the simulation runs, we aim to get some findings on how local cache affects the overall performance of the communications with respect to P(hit), cache size, popularity distribution, and number of users. All the results are discussed in Chapter V.

3. Case 2: Modeling with Local Cache and Cloudlet

Building on Case 1, we evaluate whether implementing cloudlets will further improve the performance. All the test parameters remain the same, with the exception of the calculation of the average response time. A new formula is worked out with the following considerations.

When a source node makes a request, it will search its local cache for the data. If the node cannot find the data it seeks in its local cache, it will look for the data in the cloudlet node (Figure 18). When this occurs, it will be considered a miss on the source node, and the source's local access time and miss ratio as well as the inter-ship access time are taken into consideration. Similarly, the cloudlet node will search for the requested data in its local cache, and if it does not find the data, it will have to make a request to the destination cloud server. Now, the cloudlet's local access time and miss ratio are taken into consideration, and added to the source's initial response time. In both situations, the average local access time is the same as both nodes are treated independently. Because the cloudlet node makes a request to the destination cloud server, the remote access time and the miss ratio needs to be taken into consideration into the formula. As a result, the formula becomes:

$$\zeta = \delta \times P(\text{hit}) + \{\beta + \delta + [\delta \times P(\text{hit}) + (\delta + \psi) \times P(\text{miss})]\} \times P(\text{miss}) \qquad (11)$$

where β is the inter-ship access time.

Inter-ship access time is affected by a few parameters, namely the volume of data, the data rate, the LOS distance, and etc. Due to the classification of ship-to-ship communications, we were unable to ascertain these parameters. Instead, we made an estimation using propagation delay. Typically, the maximum LOS distance between the source node and the cloudlet node is 20km. Therefore, a two-way propagation delay is calculated to be approximately 130μsec.

$$\text{Propagation\_delay} = \frac{2 \times 20000}{3 \times 10^8} \approx 130 \mu \sec$$

As compared to the local access time, this inter-ship access time is quite insignificant. So for Case 2, it is reasonable to assume that the inter-ship access time is negligible. We also assume that inter-ship communication is always available. As a result, the formula is reduced to:

$$\zeta = \delta \times P(\text{hit}) + \{\delta + [\delta \times P(\text{hit}) + (\delta + \psi) \times P(\text{miss})]\} \times P(\text{miss}) \qquad (12)$$

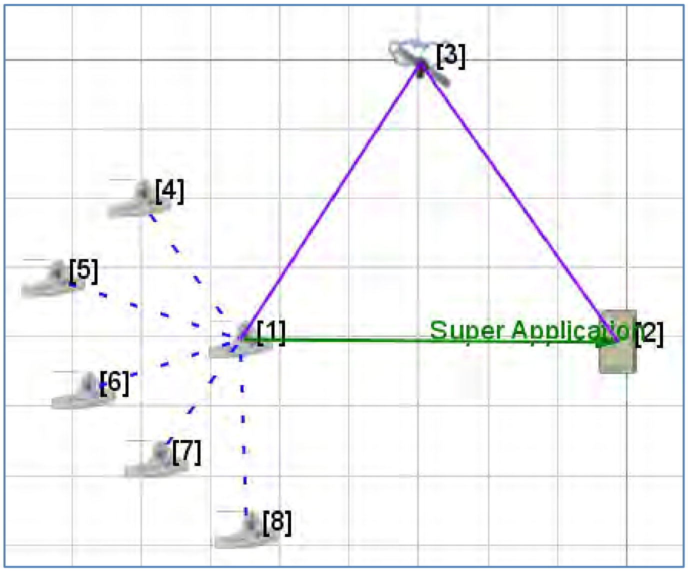


Figure 18. QualNet Model for base case with local cache and cloudlet.

The average remote access time is still dependent on TR. In this case, TR is obtained from the number of nodes connected to the cloudlet node. Let *n* be the number of connecting nodes. Effectively, by increasing the number of nodes, it is equivalent to increasing the number of users. For example, we assume that each node can support ten users. When one node is connected to the cloudlet node ($n = 1$), the TR is based on a maximum of 20 users, where ten users belong to the cloudlet node and ten users belong to the connecting node. When $n = 2$, the TR is based on a maximum of 30 users, and so on and so forth. P(hit) is taken into consideration when obtaining the number of users that contributes to the TR. When P(hit) is 0, it implies that all users from the connecting node will be making a request through the cloudlet node, resulting in the maximum number of users. The examples are illustrated as follows:

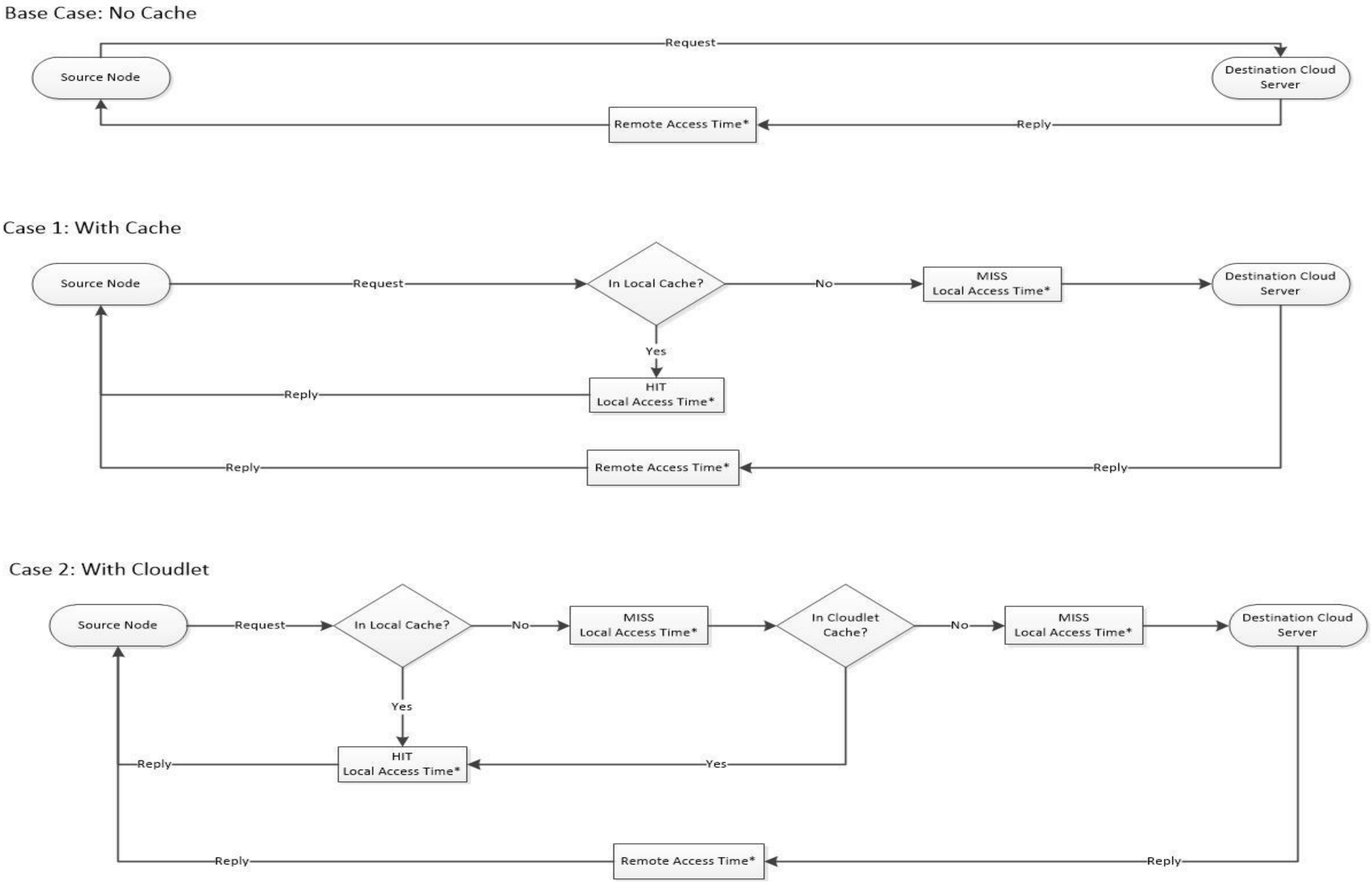
$$\text{Maximum no. of users} = 10 + 10 \times n \times (1 - \text{P(hit)})$$

$$n = 1 \rightarrow \text{Maximum no. of users} = 10 + 10 \times 1 \times (1 - \text{P(hit)})$$

$$n = 2 \rightarrow \text{Maximum no. of users} = 10 + 10 \times 2 \times (1 - \text{P(hit)})$$

Through this experiment, apart from accessing the performance provided by the cloudlet implementation, we are also able to find the optimal/maximum number of nodes that can be connected to one cloudlet so that the available bandwidth can be optimized. The results are to be discussed in the next chapter.

A summary of the three cases that we have simulated are represented in Figure 19. The three cases are shown separately, each describing the processes involved from the source node making an initial request of the data to the receipt of the reply from the remote cloud server. The method of calculating the access times is also visually explained in the flowchart. The next chapter will focus on the results that are generated from the models. A deeper analysis of the results will be conducted to help us understand the various effects when different parameters are varied. This will facilitate proper recommendations to the architecture of shipboard designs.


Base Case: No Cache

Case 1: With Cache

Case 2: With Cloudlet

Figure 19. Flow charts visually summarizing the three cases.

V. DISCUSSION OF RESULTS

In this chapter, we discuss the results and findings from the simulations that we described in Chapter IV. Our approach is to build up from the base case where there is no cache, move on to the case where there is a local cache, and then introduce cloudlets into the architecture.

The discussion is aimed to show that the proposed mitigation strategies are able to provide resolution to the identified problems in the current architecture. First, we want to show that the implementation of caching can improve the response time of requests made by the users. Second, we want to show that the implementation of cloudlets can indeed bring the “cloud closer to the users” and improve the overall performance of communications involving remote cloud server. With that, we can show that cloudlet is a feasible way to mitigate DIL environment in naval communications.

A. BASE CASE WITHOUT CACHE

In this section, the results for the base case where the naval platforms have no local cache are presented and discussed. This is the case where the naval platform has to request for the all data objects from the remote, land-based cloud.

The average response times are calculated using the average remote times collected from the QualNet. Figure 20 shows the results for average response times in seconds, plotted against a range of SATCOM bandwidth from 1 to 2.5 Megabits per second (Mbps). Six curves, representing ten to 60 users, are plotted in the same graph as shown in Figure 20. (To revisit the method of deriving data demand, refer to Figure 14.)

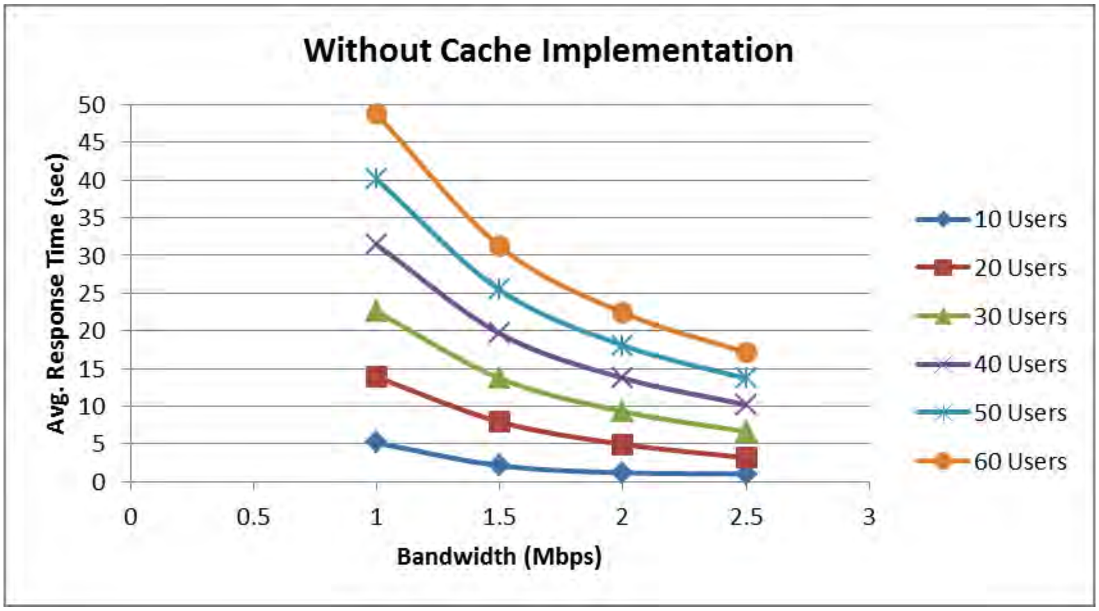


Figure 20. Average response time plots for base case (with no cache).

In Figure 20, it is observed that there is a higher rate of improvement in the average response time when the bandwidth is increased from 1 Mbps to 1.5 Mbps. When there are more users (more load to the communication channel), the improvement seems to be more obvious.

Based on the trend of the curve (left-hand side), we can infer that it is more significant to improve the bandwidth of the communications when network connectivity is limited. Intuitively, this matches with the expected behavior.

When the bandwidth increases to be higher than 2 Mbps, it is observed that the rate of improvement in the average response time becomes more gradual. For subsequent simulations, the bandwidth is fixed at 2 Mbps. This is reasonable because we can infer how bandwidth will affect the behavior of the performance by varying other parameters which are more interesting.

B. CASE 1: WITH LOCAL CACHE

In this section, we aim to see how local cache can affect the performance of the communication. This is where each naval platform has the ability to store data objects from the Internet so that the some of the data objects are available locally and there is no need to request it from the remote cloud.

1. Effect of Varying Cache Ratio

In this experiment, we vary the percentage of cacheable data and keep the other parameters, P(hit) and number of users, constant. This way, we can observe the behavior specific to percentage of cacheable data (hereafter, we refer to it as cache ratio).

In Figure 21 (a), we see multiple curves plotted on the same graph for ten users, each representing one P(hit) value. Hence, we have eight curves for P(hit), ranging from 0.1 to 0.8. Figure 21 (b) shows a similar graph, but with the number of users fixed at 60 users.

From the two graphs, we can observe that the cache ratio does not have a significant effect on average response time. For the case of 60 users, the cache ratio affects the response time slightly more but rather insignificantly. Hence, for the subsequent simulations, we would just look at 0.3 and 0.7 cache ratio so as to observe the results at the two extremes.

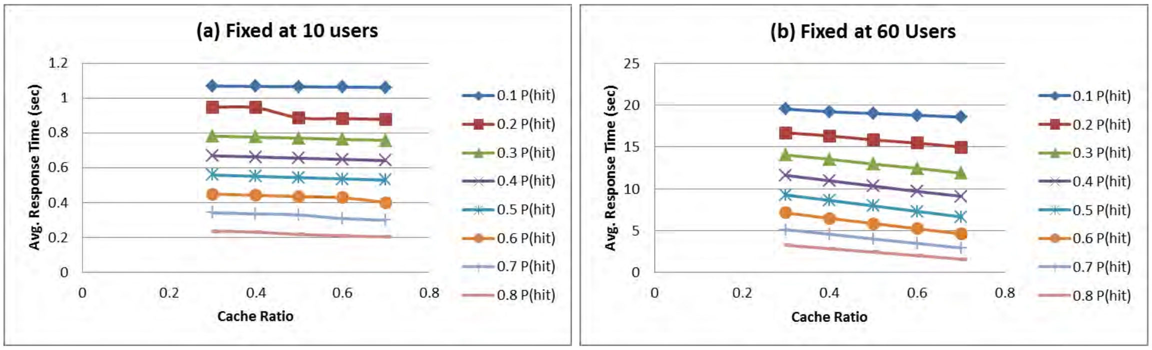


Figure 21. Average response time plots with (a) ten users, (b) 60 users.

2. Effect of Varying the Number of Users

After looking at the effect of the cache ratio, we now look at how varying the number of users will affect the performance of the data communications. As mentioned in the previous section, we fix the cache ratio at 0.3 and 0.7, while different P(hit) is represented on different curves (Figure 22).

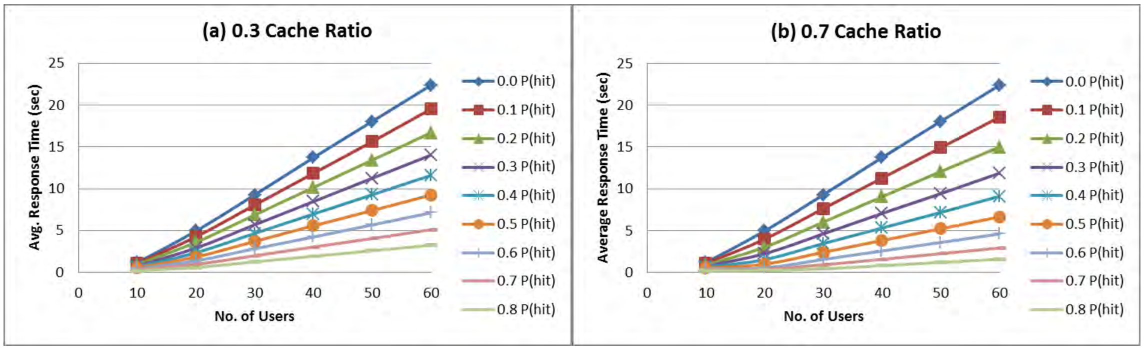


Figure 22. Varying the number of users with (a) 0.3, (b) 0.7 cache ratios.

Both (a) and (b) of Figure 22 show quite similar results, which reinforce our observations that cache ratio does not have a significant impact on the performance. Based on the graphs, we can make the following observations:

- Increasing the number of user essentially increases the data load. By comparing the curves (taking note of P(hit)=0.1 and P(hit)=0.8), it is observed that the performance of the communications is greatly affected by the number of users when P(hit) is 0.1 but not as much when P(hit) is 0.8. From this we can conclude that compared with low P(hit), a high P(hit) leads to less severe performance degradation as the volume of data (or the number of users) increases..

- Referring to Figure 16, we know that P(hit) is directly related to the cache size and popularity index (alpha). Therefore, P(hit) can be improved by increasing cache size or increasing the alpha value.

3. Effect of Varying Hit Ratio, P(hit)

Figure 23 (a) and (b) plot the same results from another perspective, where average response time is plotted against P(hit). Two separate graphs are plotted with 0.3 and 0.7 percentages of cacheable data, respectively.

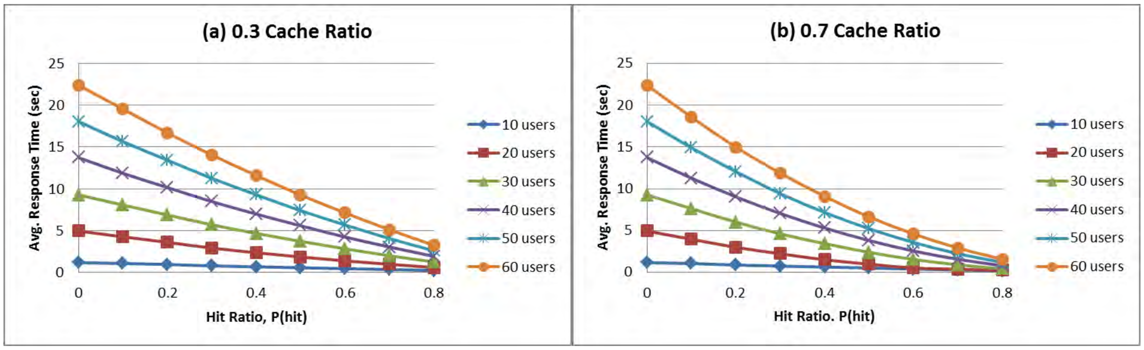


Figure 23. Varying P(hit) with (a) 0.3 (b) 0.7 cache ratios.

Although the same set of results is being used, we believe that the analysis, when P(hit) is varied should give a different trend of the performance. With (a) and (b) of Figure 23, we make the following observations:

- We have previously observed that the percentage of cacheable data does not play a big part. Here, we observe that a higher cache ratio increases the rate of performance improvement with higher P(hit); it is especially obvious in the case of 60 users.
- We also observe that higher P(hit) greatly improves the performance in the case of 60 users (higher data load) as compared to the case of ten users. This observation suggests that it is not recommended to improve the P(hit) for relatively low data load conditions if the cost of doing that is high.
- Comparing with the results from the base case, the above observations also imply that having caches can improve the overall performance of the data communications.
- The simulations were done using bandwidth fixed at 2 Mbps. It is reasonable to infer that with higher bandwidth, the curves would just shift downwards, but the trend would remain the same.

C. CASE 2: WITH CLOUDLET

In this simulation we focus on how the implementation of cloudlets will affect the performance of the data transmission. The number of users is fixed at ten for the connecting node and cloudlet in the simulations.

Reiterating our approach (refer to Chapter IV, Section C), the variable *n* is the number of nodes that are connecting to the cloudlet node at the same time. Increasing the value of n is equivalent to increasing the data demand or the data load. Similar to previous cases, the simulation is carried out with varying data loads and hit ratios. However, in this case, we assume that ship-to-ship communication is always available and the inter-ship access time is negligible. This is because we are only interested in the time required to fetch the data and not the ship's communication time. If inter-ship access time is to be taken into consideration, we foresee that this small time constant will cause the graphs of our results to exhibit a slight upward shift without changing the nature of the curves.

As before, further analysis is conducted on 0.3 and 0.7 cache ratios, which forms the lower and upper bound on the cache ratio, respectively. We compare the results for cases before and after cloudlets are implemented (shown in Figure 24 and Figure 25). The overall trend for the average response time is decreasing. This is desirable because the lower the response time, the better is the performance.

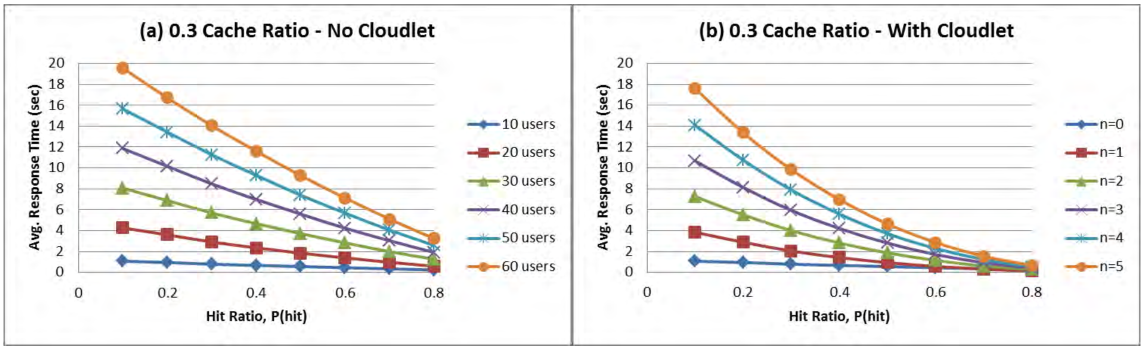


Figure 24. Average response time (with 0.3 Cache Ratio) (a) *without* cloudlet, (b) *with* cloudlet.

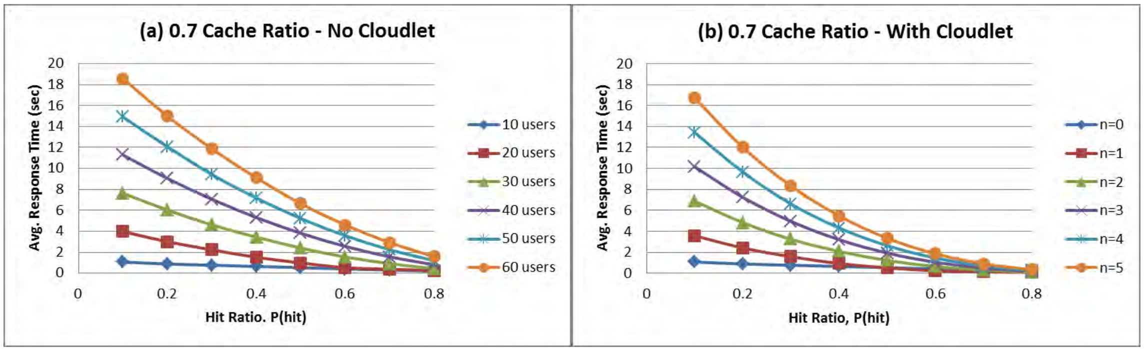


Figure 25. Average response time (with 0.7 Cache Ratio) (a) *without* cloudlet, (b) *with* cloudlet.

1. Faster Rate of Improvement in Performance with Higher P(hit)

Having more nodes connecting to the cloudlet node will cause the data load to increase proportionally on the cloudlet node. Based on the graphs (Figure 24 and Figure 25), we have the following observations:

- The slopes are steeper for the case with cloudlet as compared to the case without cloudlet.
- The improvement is more significant in the case of higher data requirements. The higher the number of nodes (*n*=5) that are connected to the cloudlet node, the better the benefits.

These observations highlight that having more nodes (indirectly more users and data loads) will not affect the performance but instead improves the performance. One possible explanation is that the when the data objects are available in the cloudlet, more nodes (*n*=5) connecting to the cloudlet would benefit from the time saved by accessing the cloudlet instead of the remote cloud.

This strengthens the point that the nodes are still able to have continuity in their operation. In a disconnected condition from the SATCOM, this provides a degraded mode of communications as compared to the direct connection via SATCOM. It is definitely better than staying in the disconnected state. However, the bottleneck caused by the cloudlet node is not studied in our research, so we cannot comment on the maximum number of nodes that can be connected to the cloudlet.

2. High Data Load Converging to Fast Performance with High P(hit)

Comparing (a) and (b) of Figure 24 and Figure 25, we have the following observation:

> The response time tends to converge at the hit ratio of 0.8. Looking at the right end of the graphs (at P(hit)=0.8), the gaps between the response times for the different number of connecting nodes (*n*=1 to 5) appear to be greater than when cloudlets are not implemented. Although this is not indicative that a 0.8 hit ratio is the optimal setting, the results further show that by implementing cloudlets we can improve the performance.

Looking from another perspective, the result is showing that the response time for a higher *n* can be as good as the response time of a lower *n* with higher P(hit). This is encouraging for the designer of the cloudlet to achieve higher P(hit) especially for the case of high data load.

D. RECOMMENDATIONS

In the base case simulations, we can see that bandwidth is an important factor in the SATCOM. It is easy to increase the bandwidth in a simulation setup so as to improve the performance, but this is not always possible in a real-world situation. Bandwidth is usually fixed or capped at a certain range and most of the time causes bottlenecks in the communications infrastructure. That is the main reason for keeping the bandwidth constant in our simulations so that we can focus our study on the cache and cloudlet.

To sum up all of the observations and findings that we have gathered, the following are the recommendations:

- It is a positive finding that cache ratio is not a major factor that is affecting the performance. Most of the time, cache ratios are not within our control, as it is depends on the nature of Internet content and the users' surfing profile.

- For the case of local cache or cloudlet, the observations tell us that if we can achieve high P(hit), we can easily increase the number of users or amount of data load, as it would not affect the performance that much. Conversely, it does not pay to improve P(hit) for relatively low data load conditions if it is not cost-efficient, due to the insignificant improvement offered.

- Introducing cloudlet can further improve the performance. It is reasonable as the nodes can request for data from the local cache of the (nearby) cloudlet instead of the much further remote cloud via SATCOM.
- In the case of DIL communication conditions involving SATCOM, introducing cloudlets to the infrastructure is a feasible mitigation.
- We recognize that the capacity of a cloudlet is limited in terms of bandwidth and cache size. This in turn limits the maximum number of end node connections to the cloudlet. This consideration can be studied in the future work.
- Overall, it is recommended to implement cache or cloudlet as it is improving the performance of naval communication.

E. CHAPTER SUMMARY

This chapter provides the analysis and discussion of the results that we have observed. In the next chapter, we will conclude the thesis and discuss the possible future works that can be extended from this research.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

Wildly fluctuating wireless bandwidth, intermittent connectivity, and unreliable connectedness (DIL connections) of SATCOM cause challenges for afloat platforms trying to connect with land-based cloud. Being able to exchange information with the cloud servers is very important to support U.S. Navy operations. As discussed in Chapter II and Chapter III, the accuracy and availability of information is vital to creating necessary situational awareness that is essential in a C2 system. Our thesis work has proposed some strategies to overcome some of these challenges.

We proposed to implement local caches and cloudlet to supplement the cloud architecture. To do a study of the proposed mitigation, we developed simulation models to conduct our experiments using QualNet, a communication network simulation application. The results and findings were then discussed. First, we were able to show that the implementation of caching can indeed improve the response time of requests made by the users. Second, we were able to show that cloudlet is able to further improve performance. The cloudlet can act as an alternative to the remote cloud when the direct connection to the satellite is down. This increases the availability of the communication network so that the operations can still move forward, although it might be in a degraded mode as compared to the direct connection via SATCOM.

Based on the initial simulation results, we are able to fix the bandwidth at 2 Mbps for subsequent simulations. This aligns with the practical situation that bandwidth is limited to a certain amount and technically challenging to improve. Subsequent simulation results suggest that cache ratio is not a crucial parameter in affecting the performance. Consequently, we fix the cache ratio in the simulations that followed.

In the next set of simulations, we are able to show that with caches and cloudlet added to the architecture, the performance can be improved. It is especially obvious in the case of using the cloudlet. Further observations from the simulations results also suggest a relationship between the number of users (directly affecting the data demand) and the hit ratio, P(hit). With high P(hit), adding more users to the network would not affect the performance as much as when we have a low P(hit). Conversely, when there are relatively more users, it is more effective to improve the P(hit).

We believe that further experiments can be planned and performed to explore additional possibilities. In the next section, the thesis report would end with a discussion of future works that can be extended from our current work.

B. FUTURE WORK

We have developed a model to demonstrate the feasibility of implementing caches and cloudlets, and show how these proposed strategies are able to improve the performance (and user experience) of communications in the mobile-cloud environment. Results obtained were positive and suggest an incentive for such implementations. However, additional work is needed to further verify the effectiveness of the strategies in real environments. Practical evaluations in the U.S Navy context are necessary, before these strategies can be put to actual use. This includes the usage of actual C2 data, as well as the integration of the inter-ship access time. This information was unavailable to us due to its sensitivity. Although we have made a reasonable assumption in Chapter IV, Section C, about the inter-ship delay being negligible, it is more complete to capture the delay in the formula for future work that follows. The inter-ship delay can be modeled dynamically with a moving naval ship. The data rate can also be modeled with the consideration of whether there is a collision medium or not.

While our thesis work examines the case where there is only one cloudlet node, the scope can be extended to study whether all nodes can take on the role of a cloudlet node. This would be analogous to establishing an ad-hoc meshed topology. Data requests can be fulfilled by searching in the local caches of all the cloudlet nodes first, before sending the requests to the remote cloud server if the data is unavailable in all the caches. This will reduce the need for a connection back to the remote cloud server via SATCOM and ensure continuity when operating in a DIL environment. In addition, optimization can be conducted to find out a few things, for example, the maximum number of users per node, the optimal number of nodes per cloudlet node, and also the maximum number of cloudlet nodes that can be supported by a certain amount of bandwidth. This will facilitate decision making in U.S. Navy operations, taking into account the tradeoffs between performance and load. We foresee that this could be achieved by tweaking the inputs and reworking the formulas used in the model that we have developed.

During our initial research on mobile-cloud related work, pre-fetching was identified as one of the solutions that could potentially mitigate latency issues. The concept behind pre-fetching is to predict the data that will be requested by users, based on a pre-determined algorithm. Accurate prediction of the data requests of users can help to pre-fetch the data from the remote cloud server and store them into local caches. This can supplement the implementation of caches, increasing the hit ratio and consequently shortening the response time. QualNet comes pre-installed with built-in libraries and it does not support the integration of pre-fetching algorithms. As a result of this limitation, we were unable to prove our claims that pre-fetching is able to contribute to an improved performance. The feasibility of this solution can be verified in subsequent research or future experimentation.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. DETAILED RESULTS FOR BASE CASE – WITHOUT CACHE

Table 9. Average Response Time for base case (*without* cache and cloudlet).

BW (Mbps)	No. of Users	Data Volume (bytes/sec)	Average Response Time (sec)
1	10	212925.6549	5.138861
	20	425851.3098	13.853641
	30	638776.9647	22.531041
	40	851702.6197	31.427041
	50	1064628.275	40.092441
	60	1277553.929	48.755141
1.5	10	212925.6549	2.105831
	20	425851.3098	7.915621
	30	638776.9647	13.700461
	40	851702.6197	19.631161
	50	1064628.275	25.407961
	60	1277553.929	31.183061
2	10	212925.6549	1.162205
	20	425851.3098	4.946681
	30	638776.9647	9.285271
	40	851702.6197	13.733321
	50	1064628.275	18.065921
	60	1277553.929	22.397121
2.5	10	212925.6549	1.037765
	20	425851.3098	3.165327
	30	638776.9647	6.636197
	40	851702.6197	10.194547
	50	1064628.275	13.660657
	60	1277553.929	17.125657

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. DETAILED RESULTS FOR CASE 1 – WITH CACHE

A. EFFECT OF VARYING CACHE RATIO ON AVERAGE RESPONSE TIME

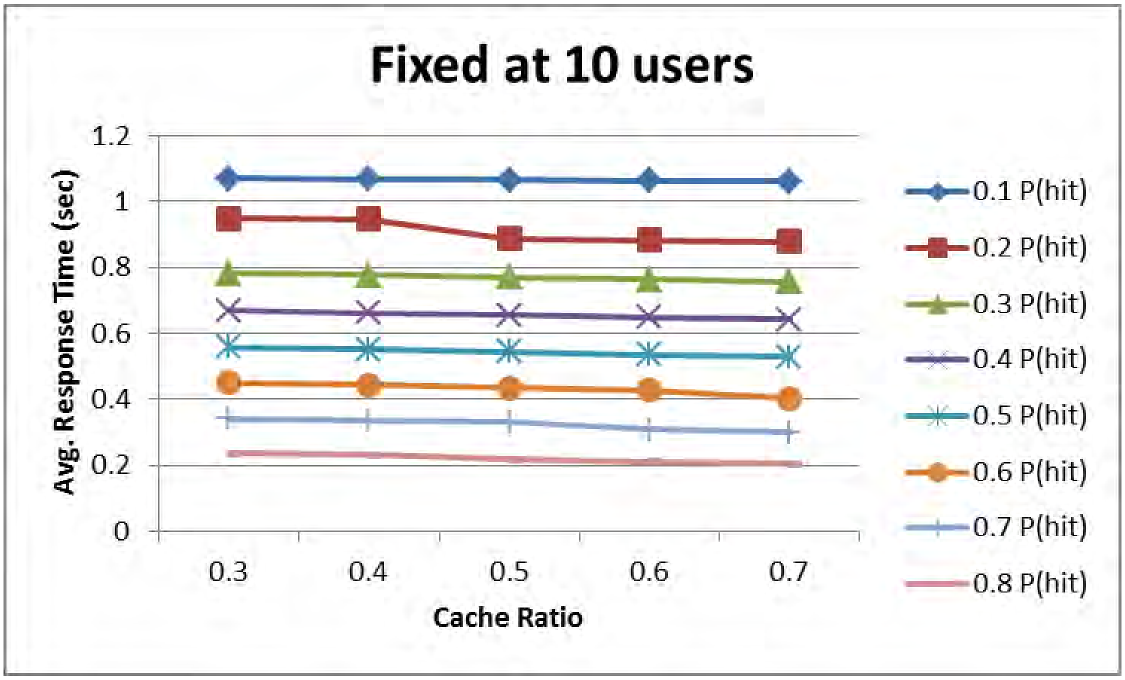


Figure 26. Fixed at 10 users.

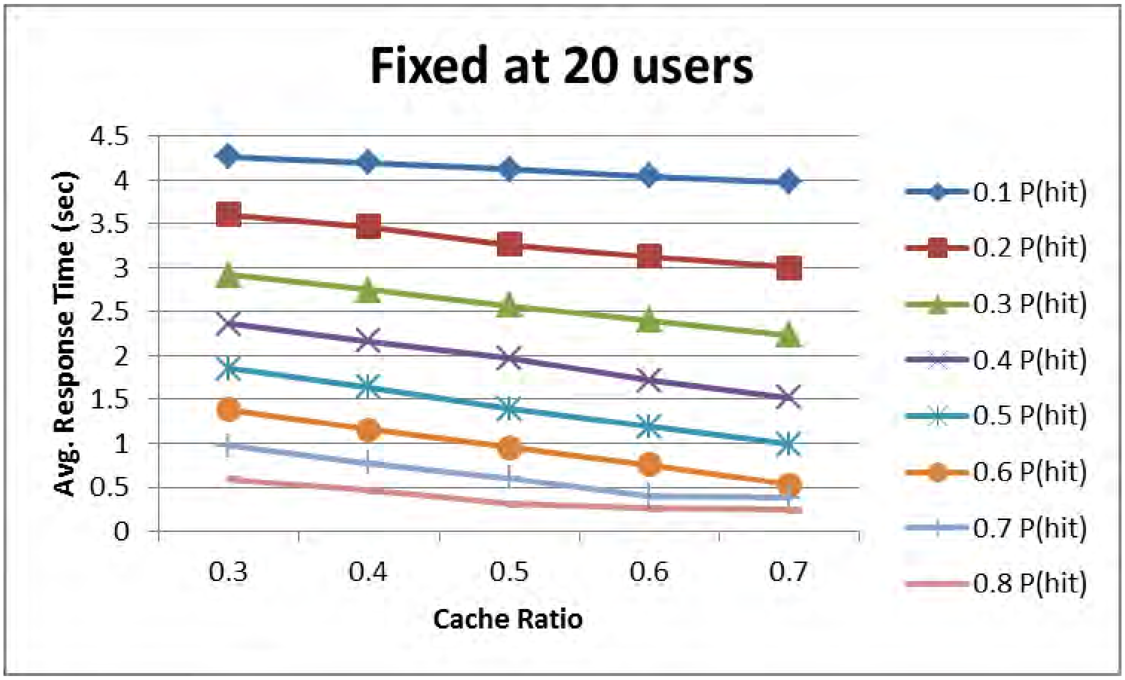


Figure 27. Fixed at 20 users.

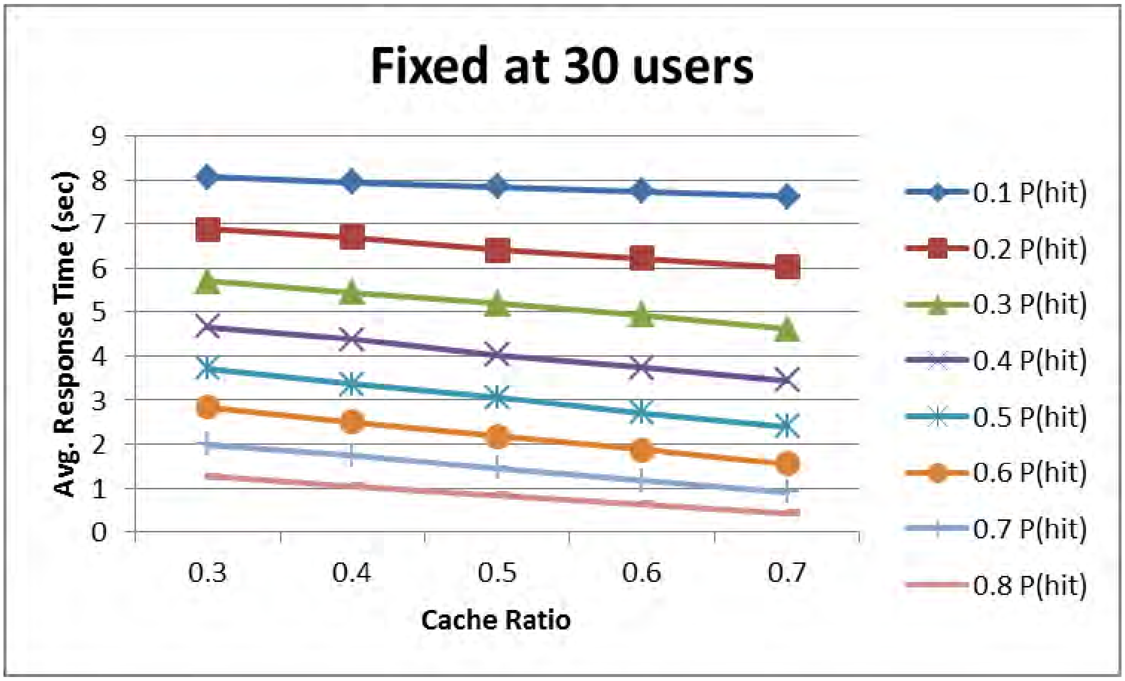


Figure 28. Fixed at 30 users.

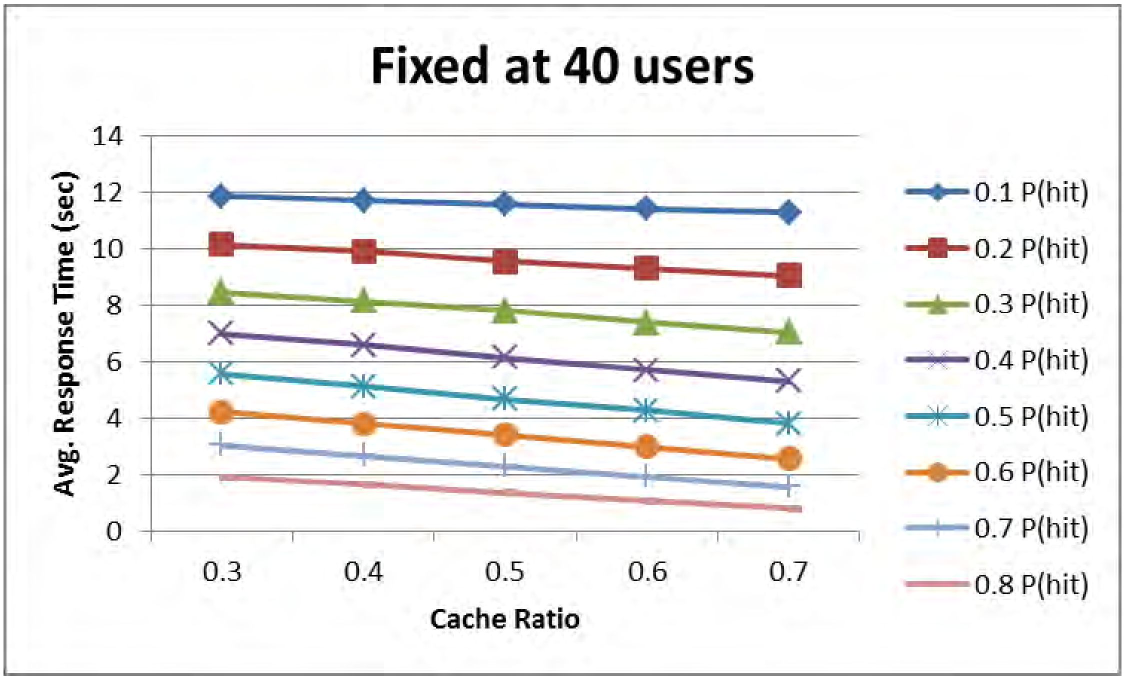


Figure 29. Fixed at 40 users.

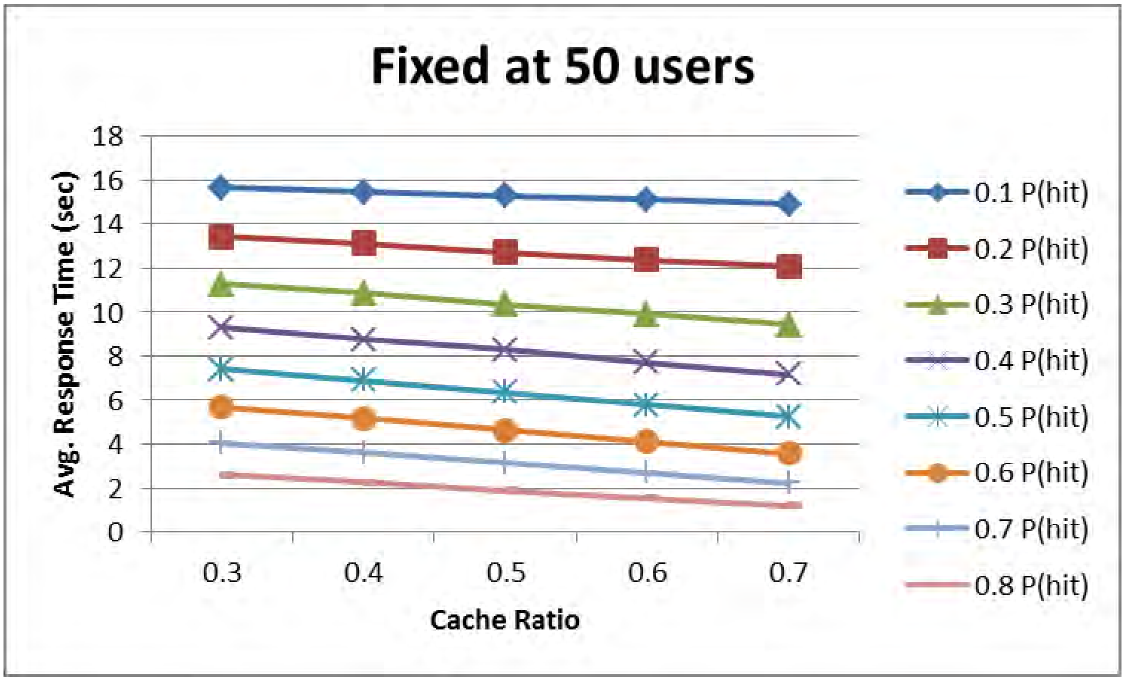


Figure 30. Fixed at 50 users.

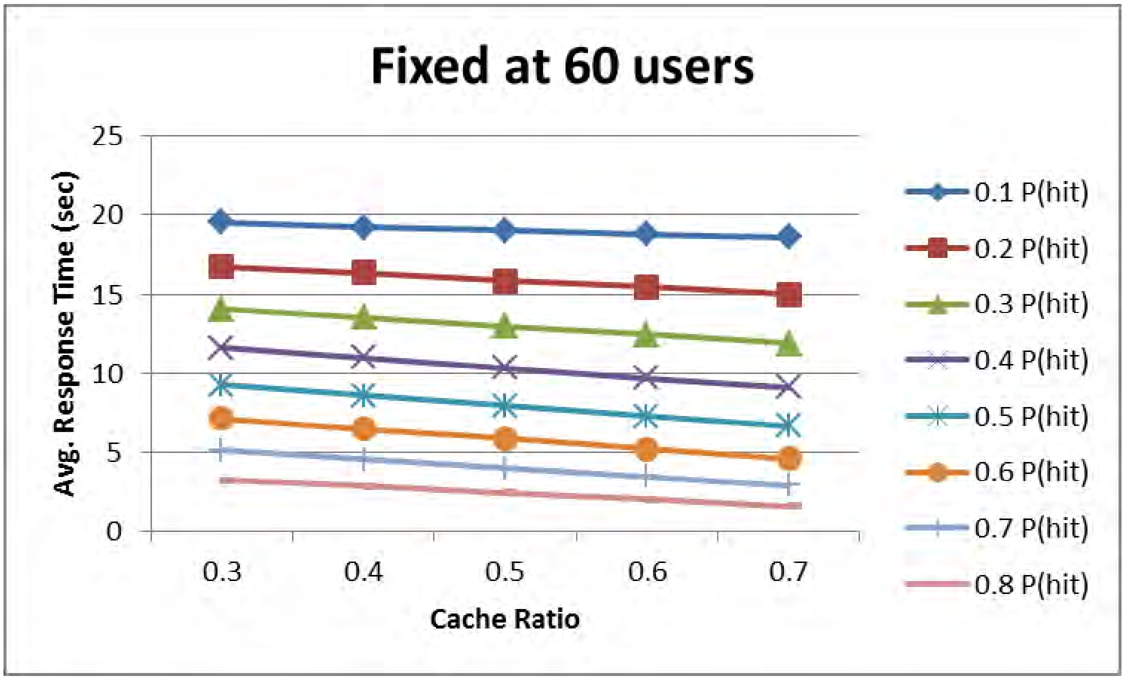


Figure 31. Fixed at 60 users.

B. EFFECT OF VARYING NUMBER OF USERS ON AVERAGE RESPONSE TIME

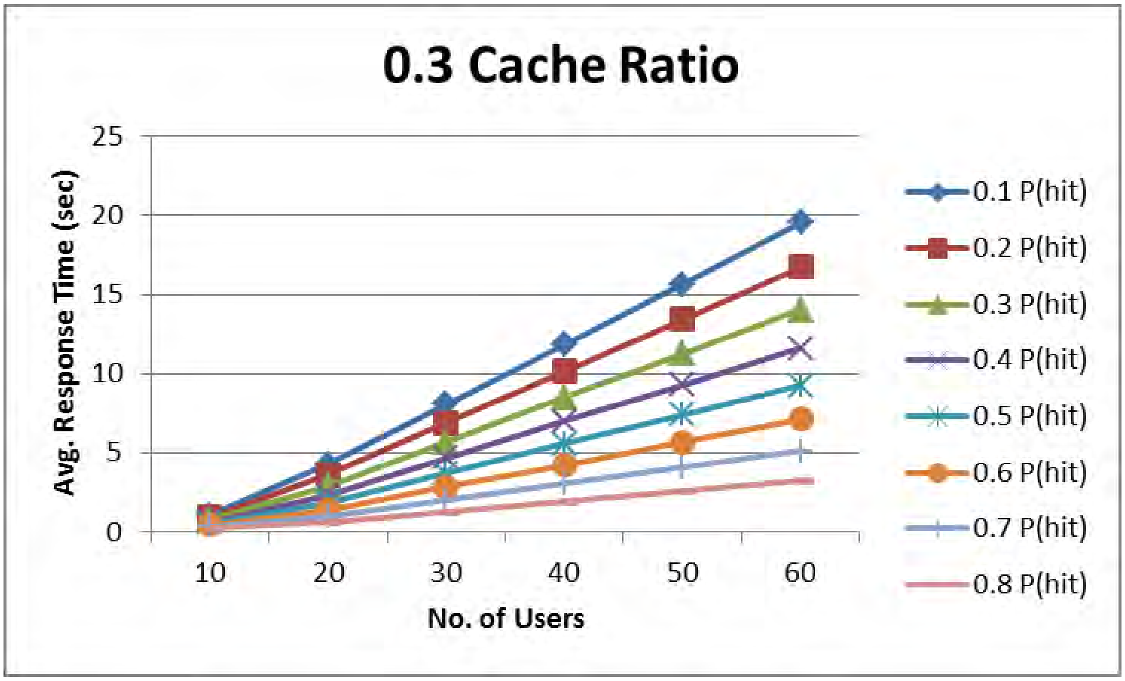


Figure 32. With 0.3 cache ratio.

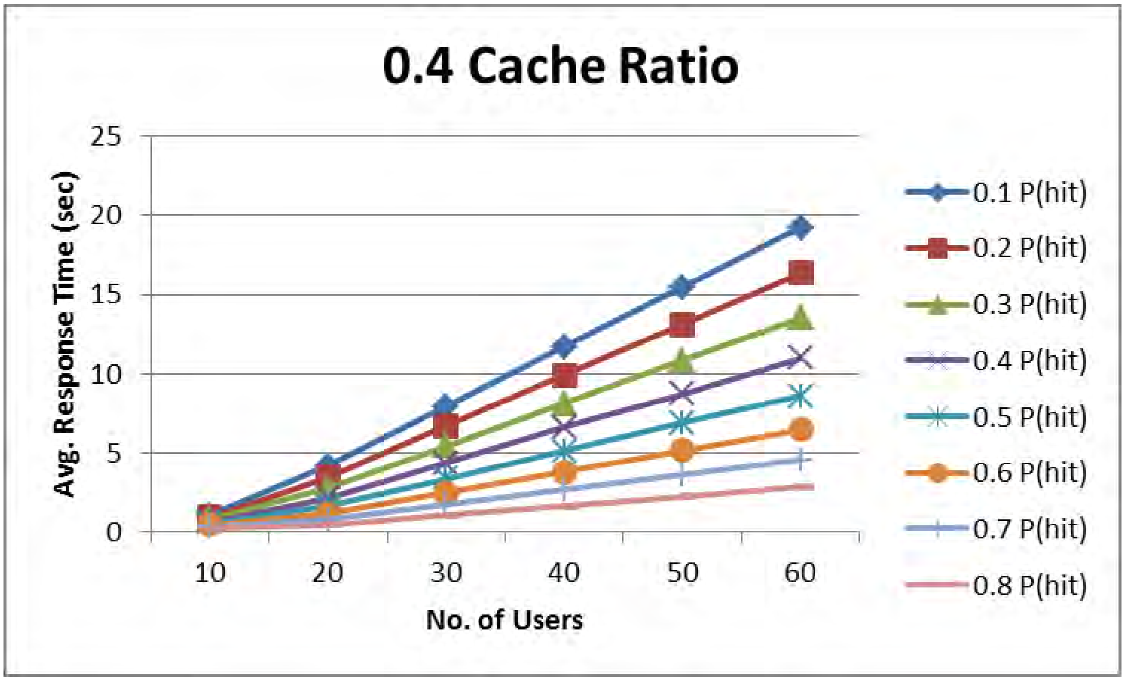


Figure 33. With 0.4 cache ratio.

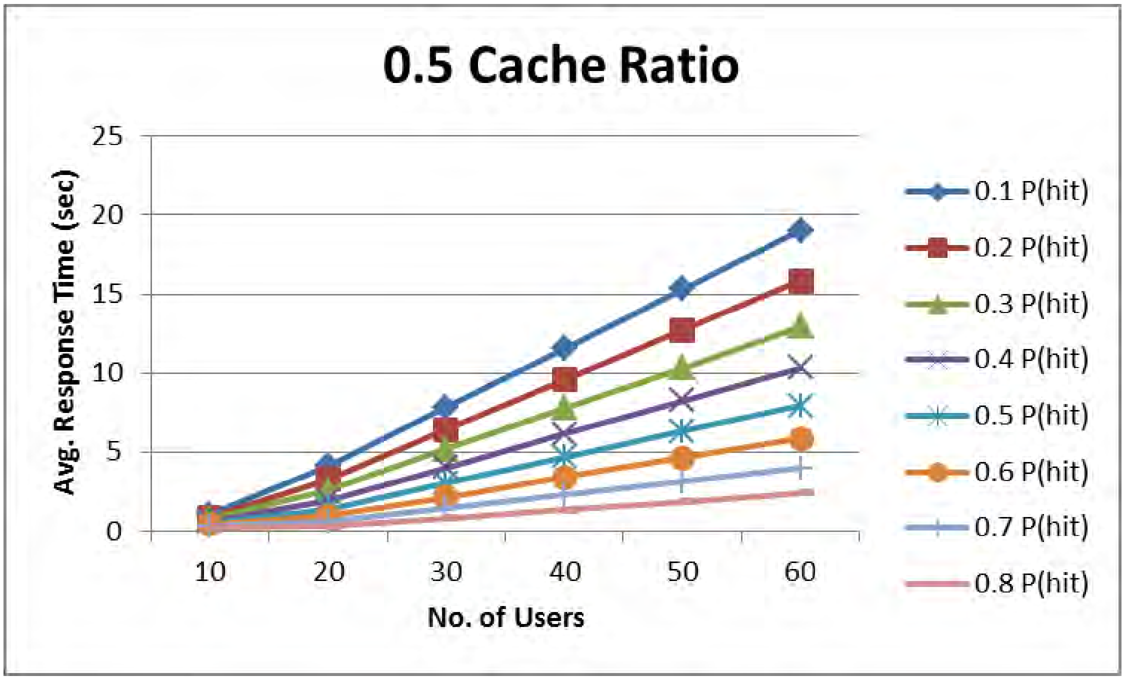


Figure 34. With 0.5 cache ratio.

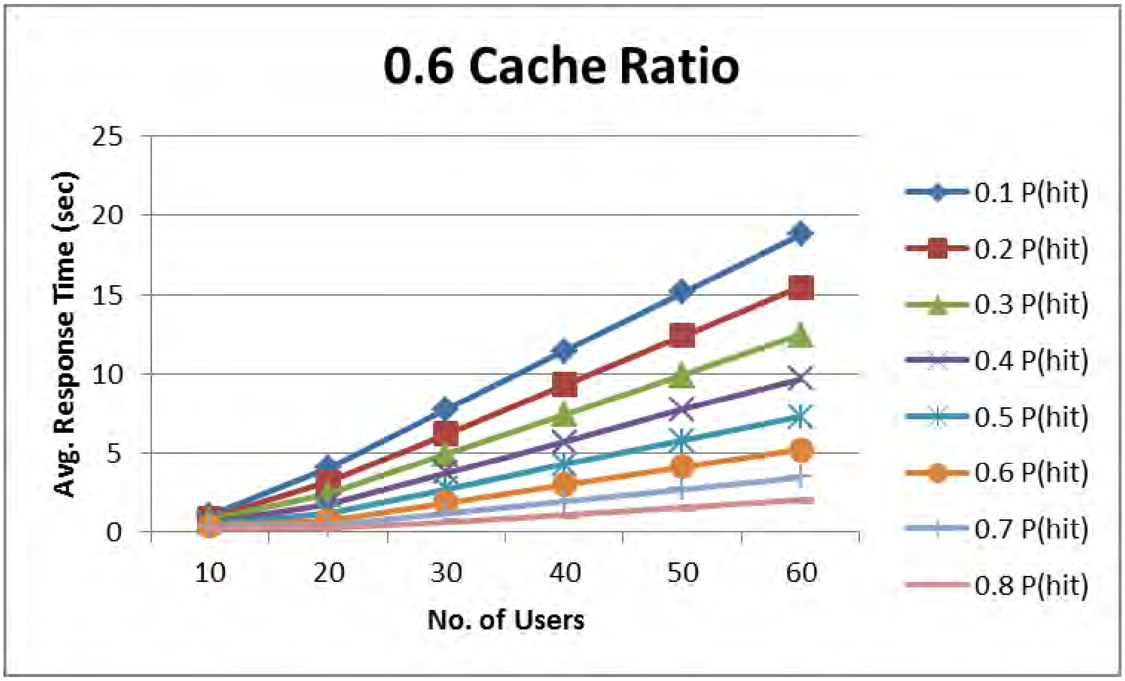


Figure 35. With 0.6 cache ratio.

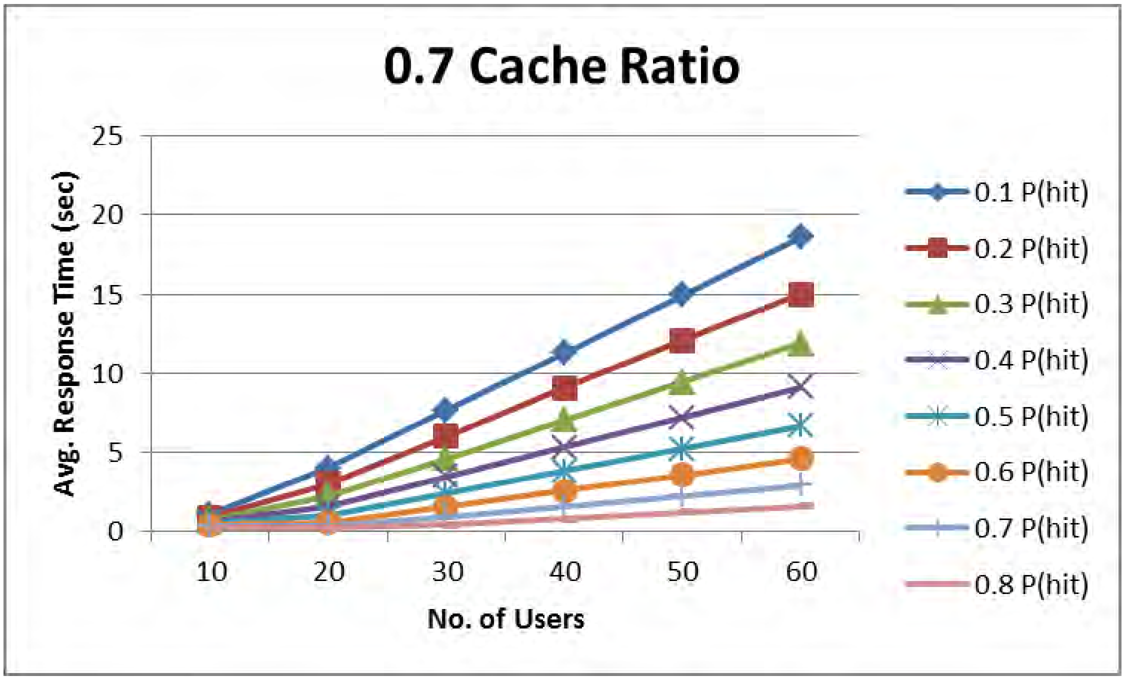


Figure 36. With 0.7 cache ratio.

C. EFFECT OF VARYING HIT RATIO ON AVERAGE RESPONSE TIME

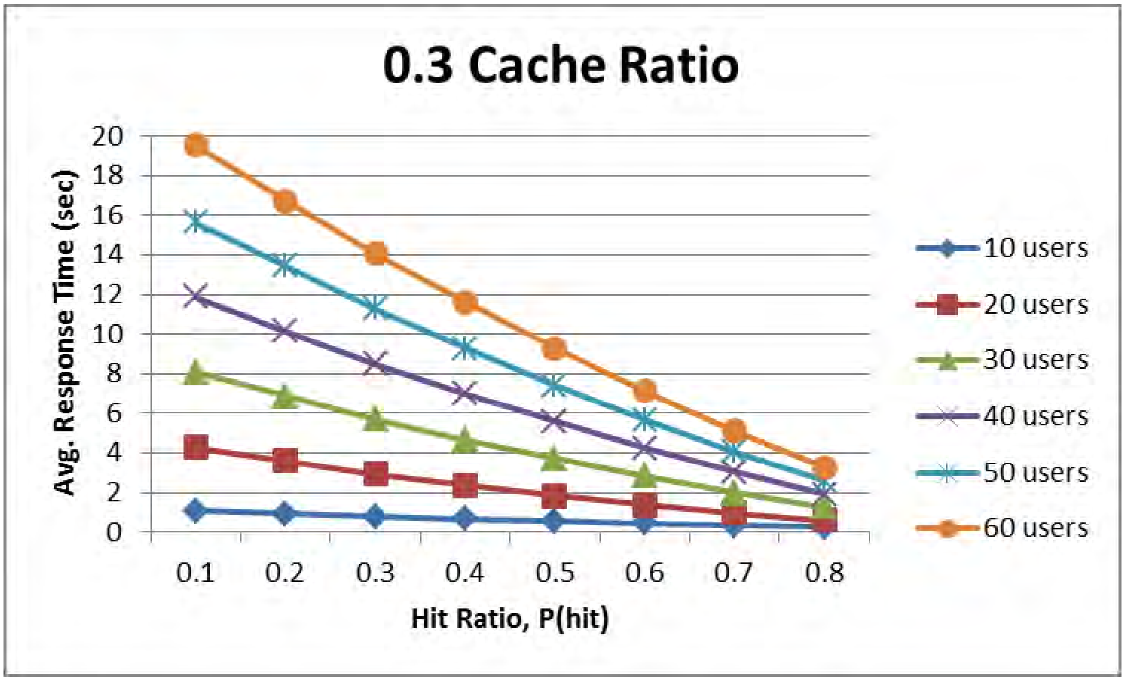


Figure 37. With 0.3 hit ratio

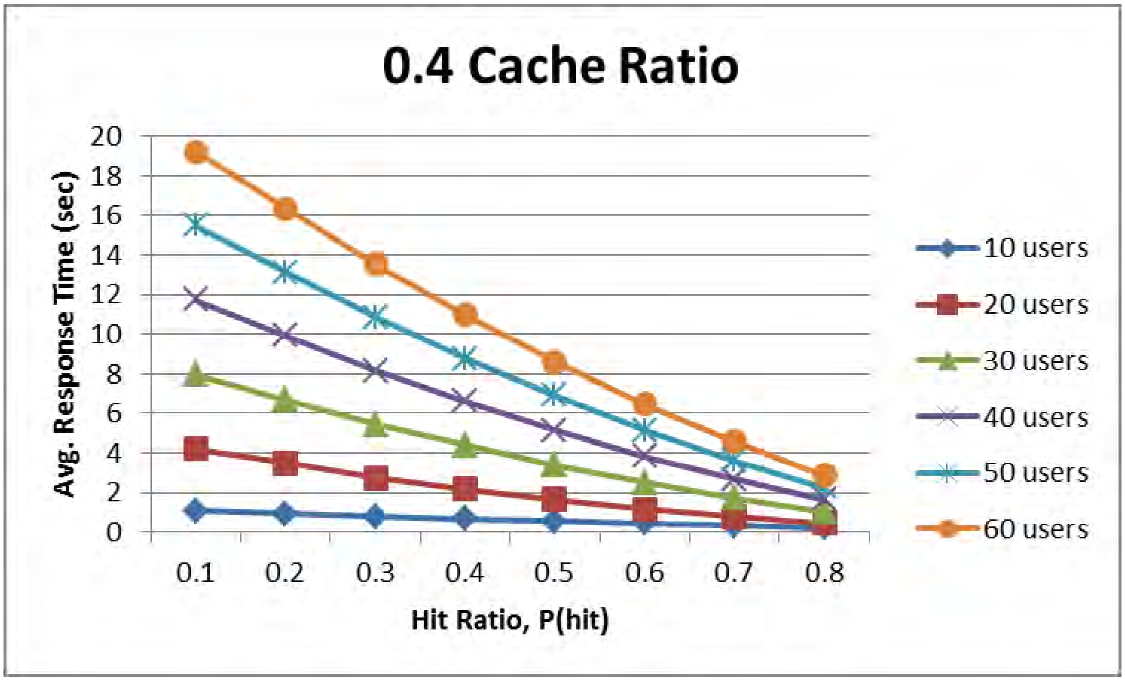


Figure 38. With 0.4 hit ratio.

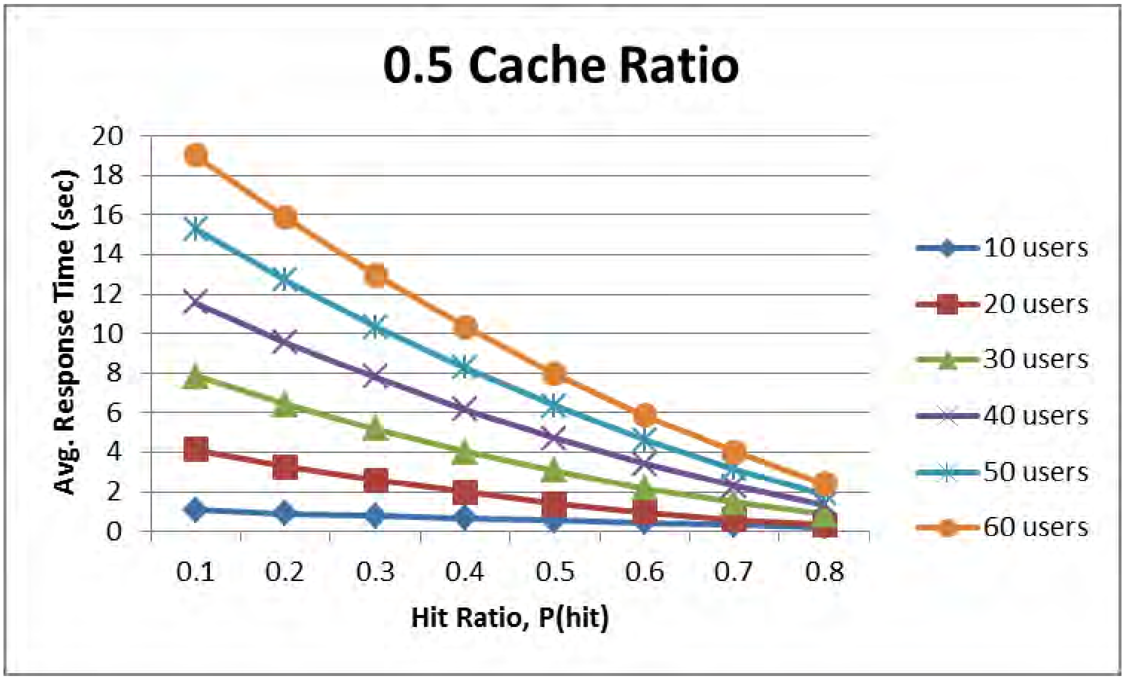


Figure 39. With 0.5 hit ratio.

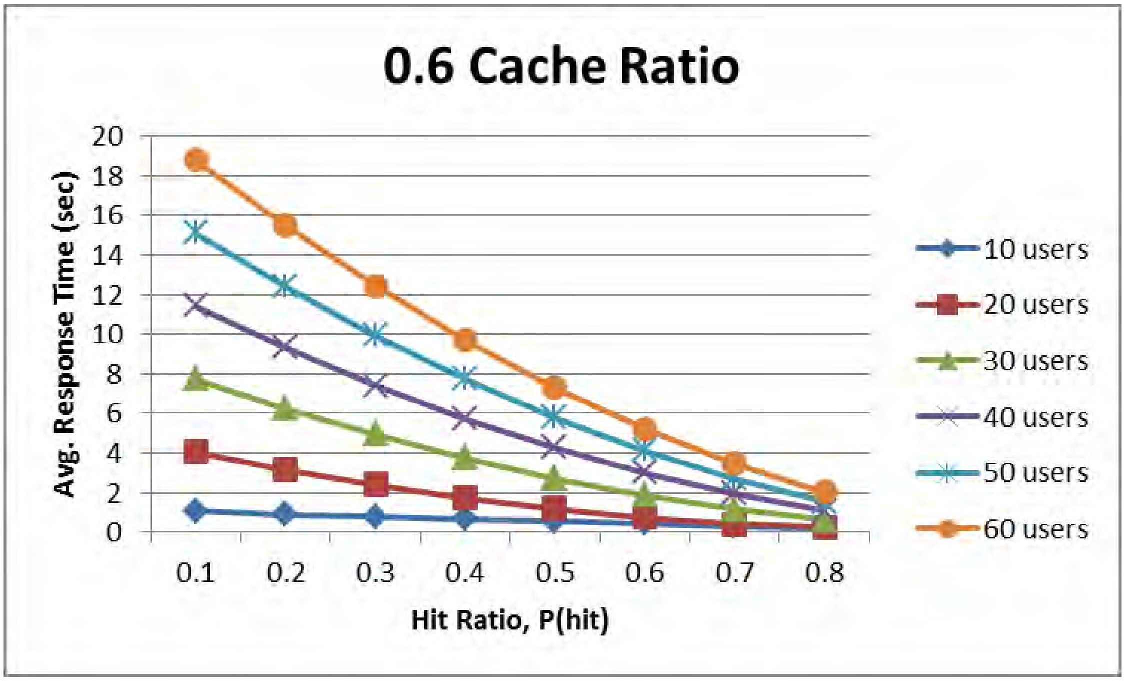


Figure 40. With 0.6 hit ratio.

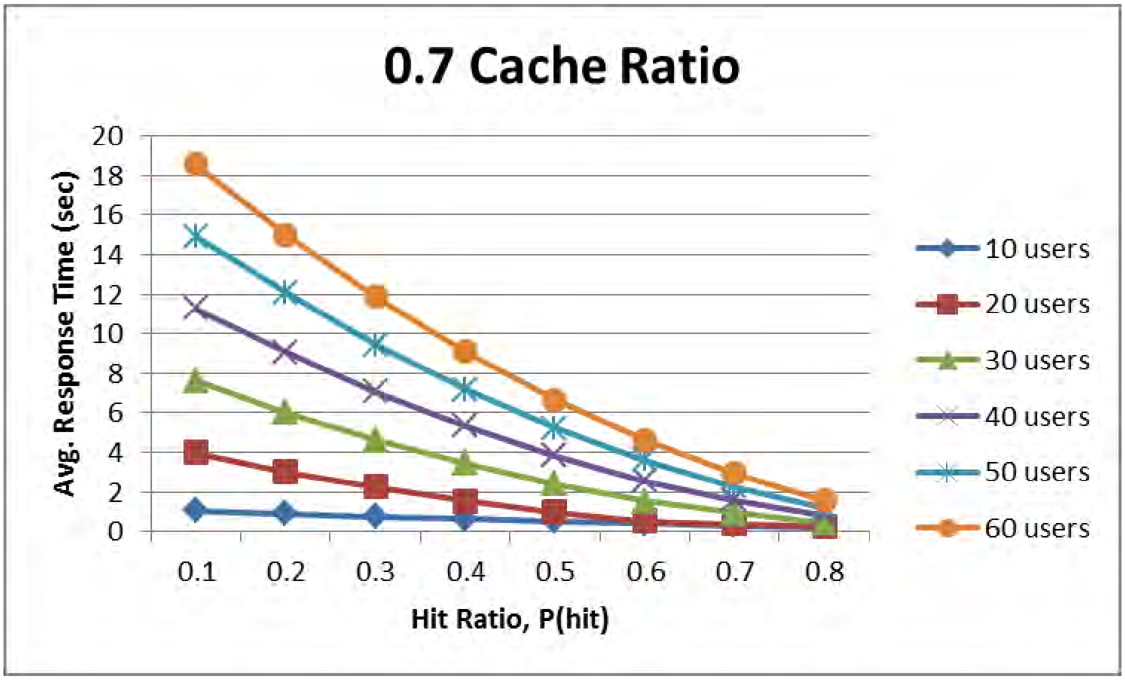


Figure 41. With 0.7 hit ratio.

APPENDIX C. DETAILED RESULTS FOR CASE 2 – WITH CLOUDLET

A. EFFECT OF VARYING HIT RATIO ON AVERAGE RESPONSE TIME

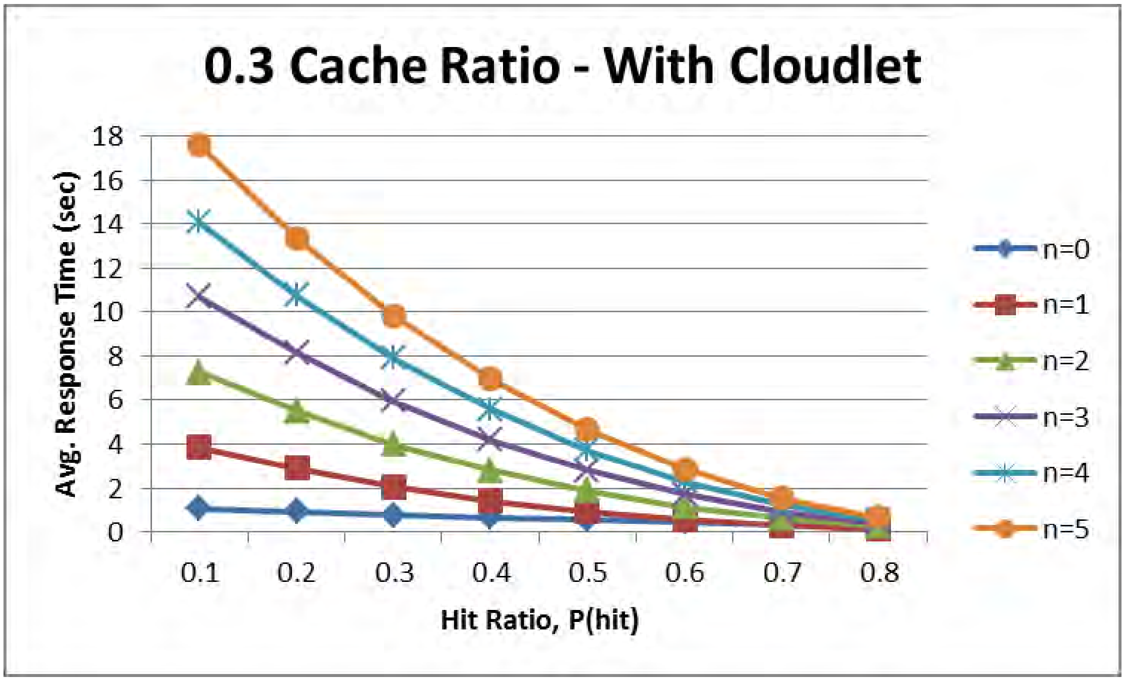


Figure 42. With 0.3 cache ratio.

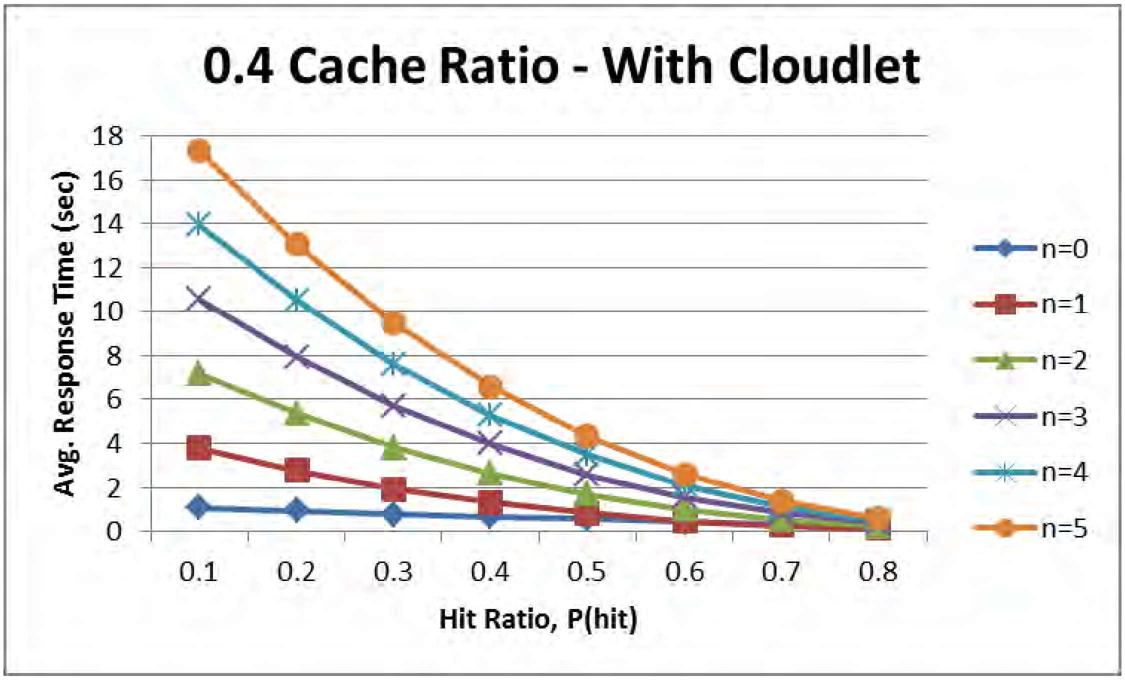


Figure 43. With 0.4 cache ratio.

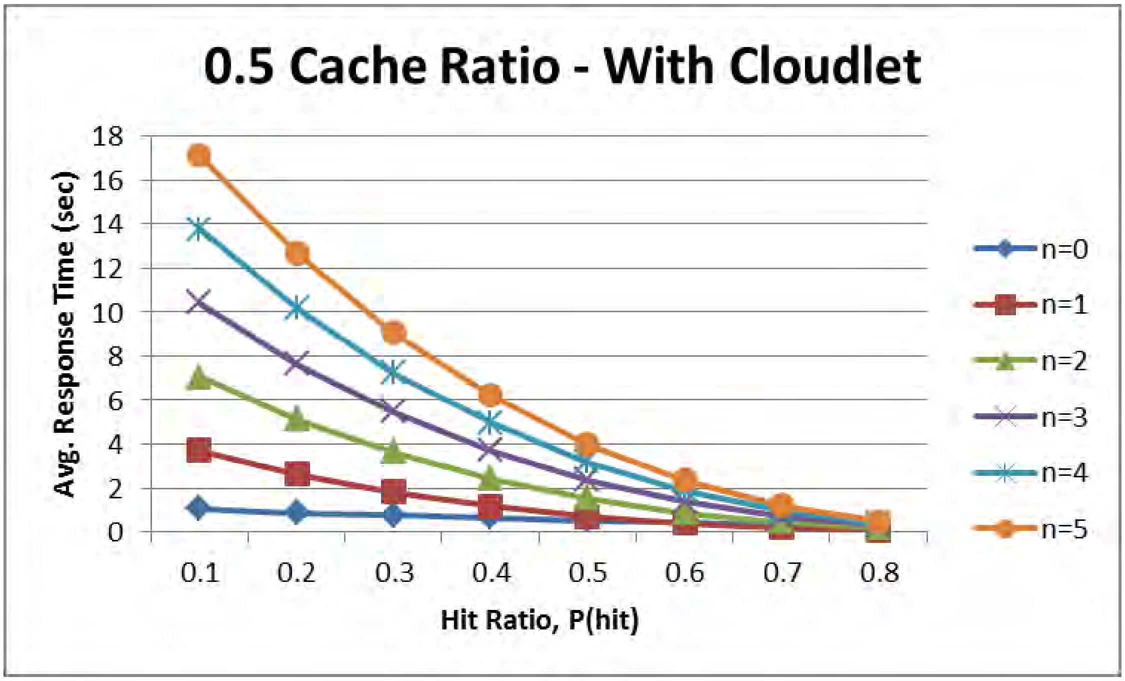


Figure 44. With 0.5 cache ratio.

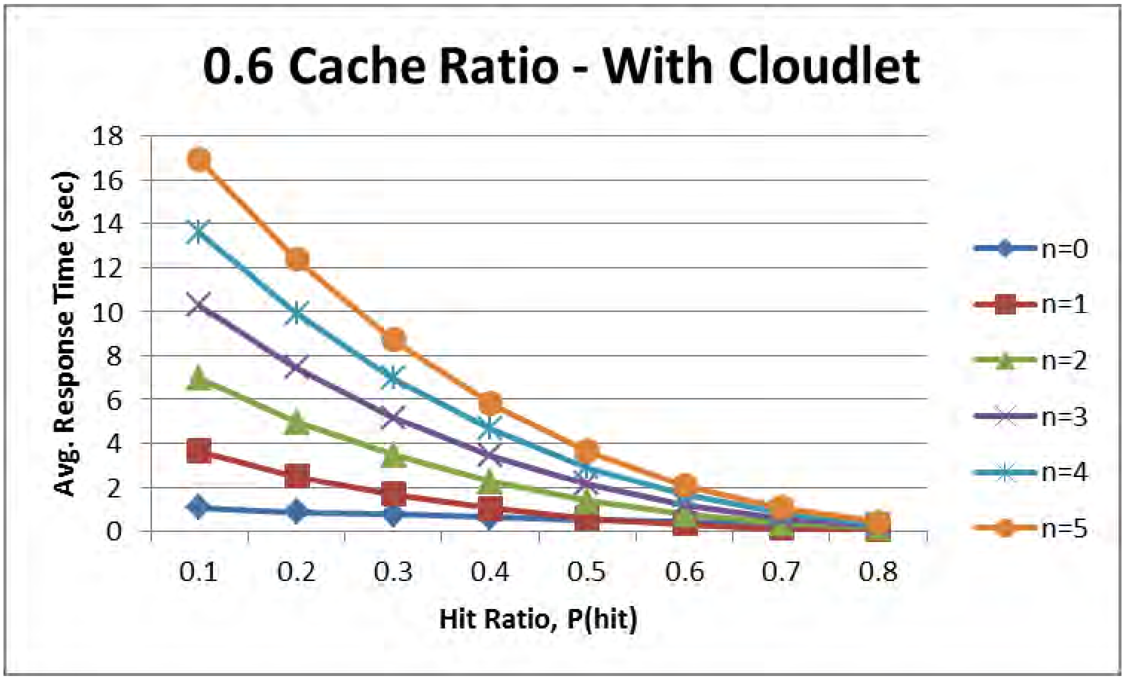


Figure 45. With 0.6 cache ratio.

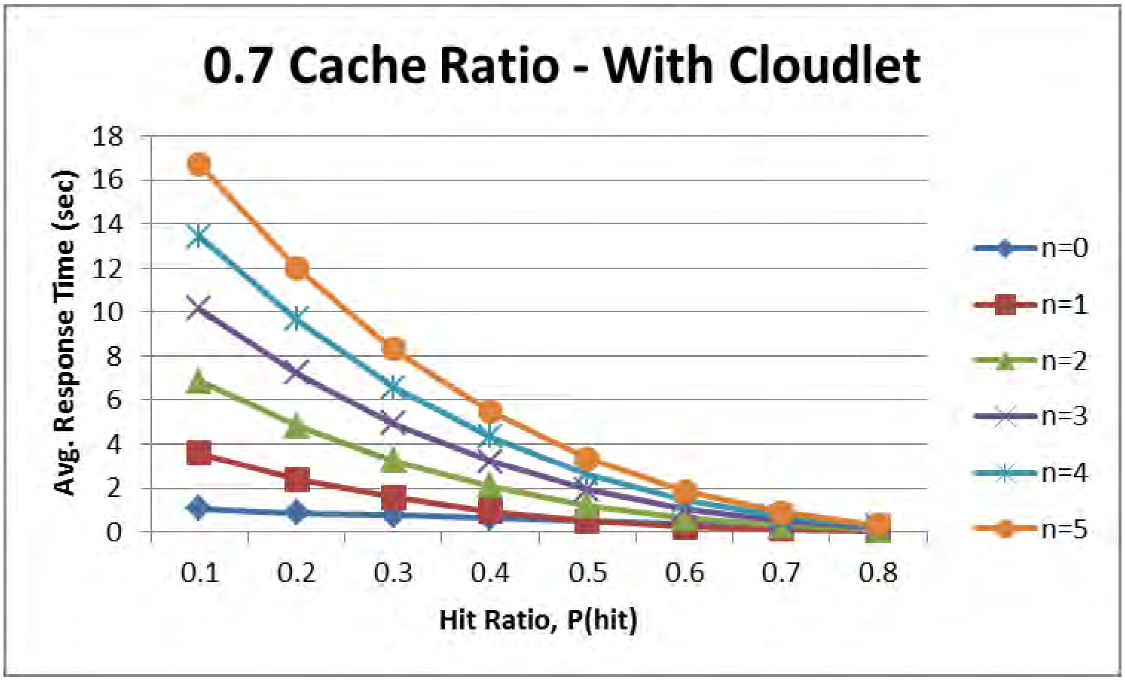


Figure 46. With 0.7 cache ratio.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Alhamad, M., Dillon, T., Wu, C. & Chang, E. (2010). *Response time for cloud computing providers*. Paper presented at WAS2010, Paris, France, November 8–10, 2013. doi:10.1145/1967486.1967579.

Beaumont, L. R. (2000, March). *Calculating web cache hit ratios*. Retrieved from http://content-networking.com/papers/web-caching.

Bekkadal, F. (2010). Novel maritime communications technologies. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, *4* (2).

Cepeda, S. (2009, August 24). *What you need to know about prefetching*. Accessed May 18, 2014. Retrieved from https://software.intel.com/en-us/blogs/2009/08/24/what-you-need-to-know-about-prefetching?language=es

Chen, H. (2013, March 20). The AWS olympics: Speed testing Amazon EC2 and S3 across regions. Accessed on June 4, 2014. Retrieved from http://www.takipiblog.com/2013/03/20/aws-olypmics-speed-testing-amazon-ec2-s3-across-regions/

Costlow, T. (2011, February 28). Commercial satellites evolve. Accessed June 3, 2014. Retrieved from http://defensesystems.com/Articles/2011/02/28/Cover-Story-Commercial-Satellites-Evolve.aspx?Page=2&p=1

Dey, S., Liu, Y., Wang, S. & Lu, Y. (2013). Addressing response time of cloud-based mobile applications. Paper presented at MobileCloud'13, Bangalore, India, July 29–August 1, 2013. doi:10.1145/2492348.2492359

FAO Corporate Document Repository. (2001). Fishing operations. Accessed on June 4, 2014. Retrieved from http://www.fao.org/docrep/003/w9633e /w9633e09.htm#bm09.2

Fesehaye, D., Gao, Y., Nahrstedt, K., & Wang, G. (2012, September). Impact of cloudlets on interactive mobile cloud applications. In Proceedings of 2012 IEEE 16th international enterprise distributed object computing conference (EDOC). Piscataway, NJ: IEEE.

Foster, K. D., Shea, J. J., Michael, J. B., Otani, T. W., Peitso, L., & Shing, M. T. (2010, August). Cloud computing for large-scale weapon systems. In Proceedings of 2010 IEEE international conference on granular computing (GrC). Piscataway, NJ: IEEE.

Greenert, J. W. (2014). *The United States Navy Arctic Roadmap for 2014 to 2030*. Washington, DC: Office of the Chief of Naval Operations.

Grumman, N. (2013). *Understanding voice and data link networking*. Accessed on May 9, 2014. Retrieved from http://www.northropgrumman.com /Capabilities/DataLinkProcessingAndManagement/Documents/Understanding_Voice+Data_Link_Networking.pdf

Hennessy, J. L., & Patterson, D. A. (2012). *Computer architecture: a quantitative approach*. Waltham, MA: Elsevier.

Huth, A. (2011). *The basics of cloud computing*. Accessed on February 10, 2014. Retrieved from http://www.us-cert.gov/reading_room/USCERT-CloudComputingHuthCebula.pdf

Khan, K. A., Wang, Q., & Grecos, C. (2012). Experimental framework of integrated cloudlets and wireless mesh networks. In *Proceedings of 20th telecommunications forum (TELFOR)*. Piscataway, NJ: IEEE.

Kung, H. T., Lin, C. K., Vlah, D., & Scorza, G. B. (2010, October). Speculative pipelining for compute cloud programming. In *Proceedings of IEEE military communications conference, 2010-MILCOM*. Piscataway, NJ: IEEE.

Lapic, S., Ching, D., Labbe, I., Jordan, M., Meagher, C., Chong, L., Dumoulin, S. & Thompson, R. (2013). *Maritime operations in disconnected, intermittent, and low-bandwidth environments*. San Diego, CA: Space and Naval Warfare Systems Center Pacific.

le Roux, W. H. (2008, June). Interoperability requirements for a south african joint command and control test facility. In *Society for Modeling & Simulation International – Proceedings of the 2008 summer computer simulation conference*. Vista, CA: ACM.

Lee, H. K., An, B. S., & Kim, E. J. (2009, July). Adaptive prefetching scheme using Web log mining in cluster-based web systems. In *Proceedings of ICWS 2009: IEEE international conference on web services*. Piscataway, NJ: IEEE.

Li, A., Yang, X., Kandula, S., & Zhang, M. (2010, November). CloudComp: Comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. New York: ACM

Li, J., & Wu, S. (2012, August). Real-time data prefetching algorithm based on sequential pattern mining in cloud dnvironment. In *2012 international conference on industrial control and electronics engineering (ICICEE)*. Piscataway, NJ: IEEE.

Li, J., Bu, K., Liu, X., & Xiao, B. (2013, August). ENDA: embracing network inconsistency for dynamic application offloading in mobile cloud computing. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. New York: ACM.

Liu, F., Shu, P., Jin, H., Ding, L., Yu, J., Niu, D., & Li, B. (2013). Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications. *IEEE Wireless Communications, 20*(3).

Michael, J. B. (2011). *Leveraging the cloud to support communications in the tactical environment*. Accessed on February 12, 2014. Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a555277.pdf

Military.com (2005). *Global broadcast system*. Accessed on May 11, 2014. Retrieved from http://www.military.com/ContentFiles/techtv_update_global.htm

OMNeT++ (2013). *OMNeT++ Network simulation framework*. Accessed on June 20, 2014. Retrieved from http://www.omnetpp.org/

Osa, J., Castello, R., Radulovich, J., Gillcrist, B., & Finocchiaro, C. (2004, April). Distributed positioning, navigation and timing (DPNT). In *Proceedings of IEEE 2004 position location and navigation symposium*. Piscataway, NJ: IEEE.

Pan, B., Wang, X., Hong, C. P., & Kim, S. D. (2012, October). AMVP-Cloud: A framework of adaptive mobile video streaming and user behavior oriented video pre-fetching in the clouds. In *Proceedings of 12th international computer and information technology (CIT).* Piscataway, NJ: IEEE.

Perkins, R., Dejesus, F., Durham, J., Hastings, R., & McDonnell, J. (2013, June). *C2 data synchronization in disconnected, intermittent, and low-bandwidth (DIL) environments*. Paper presented at 18th ICCRTS, June 19–21, 2013, Alexandria, VA.

Sandvine (2014). *Global Internet phenomena report 1H2014*. Retrieved from https://sandvine.com/downloads/general/global-Internet-phenomena/2014/1h-2014-global-Internet-phenomena-report.pdf

Satynarayanan, M. (n.d.). Elijah - Cloudlet-based mobile computing. *Carnegie Mellon University – Elijah homepage.*
Accessed on March 2014. Retrieved from http://elijah.cs.cmu.edu/.

Satyanarayanan, M., Bahl, P., Caceres, R., & Davies, N. (2009). The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, *8*(4), 14–23.

Satyanarayanan, M., Lewis, G., Morris, E., Simanta, S., Boleng, J., & Ha, K. (2013). The Role of Cloudlets in Hostile Environments. *IEEE Pervasive Computing*, *12*(4), 40–49.

Scalable Network Technologies, Inc. (2013). *QualNet 7.1 programmer's guide*. Culver City, CA: Scalable Network Technologies.

ScaleOut. (2007). *Distributed caching: Fast response time and scalability*. Retrieved from http://h21007.www2.hp.com/portal/download/product/24028/SOSS_Performance_06-07_1207761260469.pdf.

Shaw Sr, P. T., Peaslee, S., & Ferguson, M. O. (2004, November). Integrated and distributed position navigation and timing (PNT) data in shipboard environments. In *Proceedings of MTS/IEEE techno-ocean '04*. Piscataway, NJ: IEEE.

Swick, M., White, J., & Masters, M. (1999). A summary of communication middleware requirements for advanced shipboard computing systems. In *Proceedings of the IEEE fifth real-time technology and applications symposium, 1999*. Piscataway, NJ: IEEE.

Waltz, E. L., & Buede, D. M. (1986). Data fusion and decision support for command and control. *IEEE Transactions on Systems, Man and Cybernetics, 16*(6), 865–879.

Wang, X., Kwon, T. T., Choi, Y., Wang, H., & Liu, J. (2013). Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users. *IEEE Wireless Communications, 20*(3), 72–79.

Wessels, D. (2001). *Web caching*. Sebastopol, CA: O'Reilly Media.

Wilson, J. W., Stein, J., Munjal, S., & Dwivedi, A. (2005, October). Capacity planning to meet QoS requirements of joint battle management and command and control (BMC2) applications. In *Proceedings of IEEE military communications (MILCOM) conference, 2005.* Piscataway, NJ: IEEE.

Yoon, U. K., Kim, H. J., & Chang, J. Y. (2010, August). Intelligent data prefetching for hybrid flash-disk storage using sequential pattern mining technique. In *Proceedings of 2010 IEEE/ACIS 9th international conference on computer and information science (ICIS).* Piscataway, NJ: IEEE.

Zhang, B., Ross, B., Tripathi, S., Batra, S., & Kosar, T. (2013, November). Network-aware data caching and prefetching for cloud-hosted metadata retrieval. In *Proceedings of the ACM third international workshop on network-aware data management*. New York: ACM.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California